Departamento de Bioingeniería e Ingeniería Aeroespacial

PROYECTO FIN DE CARRERA

DESARROLLO DE SOFTWARE DE LIBRERIAS Y APLICACIONES PARA EL CONTROL DE SISTEMAS MICRO-CT

Autor: Ana Ortega Gil

Tutor: Alejandro Sisniega Crespo

Leganés, 18 de julio de 2013

Título: **DESARROLLO DE SOFTWARE DE LIBRERIAS Y APLICACIONES PARA EL CONTROL DE SISTEMAS MICRO-CT**
Autor: **Ana Ortega Gil**
Director: **Alejandro Sisniega Crespo**

EL TRIBUNAL

Presidente: Manuel Desco Menéndez

Vocal: Julio Villena Román

Secretario: Javier Pascau González Garzón

Realizado el acto de defensa y lectura del Proyecto Fin de Carrera el día 18 de julio de 2013 en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de

VOCAL

SECRETARIO PRESIDENTE

Agradecimientos

El proyecto fin de carrera es el último proyecto que se elige exclusivamente por y para uno mismo. Es el momento de disfrutar con lo que has aprendido.
Agradezco al Laboratorio de Imagen Médica del Hospital Gregorio Marañón, en especial a M. Desco y a J.J Vaquero, que me brindasen la oportunidad y la idea para realizar este proyecto.
A mi guía y modelo, A. Sisniega, por estar siempre disponible para resolver mis dudas, corregir mis errores y compartir sus conocimientos. Ha conseguido contagiar a sus *proyectandos* su motivación y dedicación por la investigación.
A los compañeros del Lim, con los que no sólo es un placer trabajar sino que es con los que apetece desconectar al terminar la jornada. Hay trabajos que poco menos que invitan a dejar la sonrisa en la puerta y recogerla a la salida; y luego está este, en el que cómo entres sin ella, te dibujan una antes de que te des cuenta.
A mi familia, gracias por vuestro apoyo incondicional, por vuestros consejos, por vuestra ayuda constante, por vuestras palabras de ánimo y por vuestra gran paciencia. Sin vosotros llegar hasta aquí hubiera sido imposible.
A mis amigos y compañeros de universidad. A los amigos de siempre. A todos con los que he compartido experiencias.

Gracias.

Abstract

Computed tomography (CT) is one of the techniques most commonly used to retrieve anatomical information of living subjects non-invasively. During the last years, the advent of x-ray Cone Beam CT –CBCT– has enabled the development of dedicated imaging systems for specific environments. Nowadays, these systems are widely used in radiotherapy and image-guided surgery, imaging of specific regions and preclinical imaging in small animals.
Despite the advance experienced by CBCT during the last decade, the differences with conventional CT make it a subject of vigorous research, mainly due to the issues introduced by the use of a larger cone aperture and two-dimensional detectors. Micro-CT systems for small-animal imaging add to the commented issues the need of a higher image resolution to observe anatomical structures with the same level of detail obtained for clinical CT.
The purpose of this project is the development of a set of low-level software libraries to control hardware elements of high resolution micro-CT systems based on microfocus x-ray sources and flat-panel detectors. The implemented libraries must offer an interface independent of the underlying hardware and the possibility of adding new hardware elements with no modifications in the high-level applications.
The designed libraries consisted of a set of individual modules controlling x-ray detectors, x-ray sources and control systems for motor drives, providing a transparent interface to the high level application. The low-level libraries were integrated into a high-level application for the acquisition and basic processing of micro-CT image.
The individual components were tested on two existent micro-CT systems, assessing the functionality of the libraries. The complete software architecture was included in an existent micro-CT system and a set of CT data was acquired and reconstructed, proving the validity of the developed software for the successful generation of CT volumes.

Keywords: x-ray, CT, micro-CT, frames, images, acquisition, C++.

Resumen

La Tomografía Axial Computarizada (TAC) es una de las técnicas más ampliamente utilizadas para la obtención de información anatómica en vivo y de forma no invasiva. Durante los últimos años, la aparición del TAC de rayos X de haz cónico –CBCT en sus siglas inglesas– ha hecho posible el desarrollo de sistemas de imagen dedicados para entornos de aplicación específicos. Hoy en día este tipo de sistemas son ampliamente utilizados en radioterapia y cirugía guiada por imagen, imagen de regiones específicas, e imagen preclínica de pequeño animal.

A pesar del gran avance experimentado por la CBCT en la última década, las diferencias con los sistemas de TAC convencionales hacen de esta técnica un área con un gran potencial para la investigación, especialmente debido a las complicaciones introducidas por el uso de un ángulo de cono mayor y por el uso de detectores bidimensionales. En el caso de sistemas micro-TAC para imagen preclínica a las complicaciones anteriormente presentadas se añade la necesidad de obtener una mayor resolución que permita la observación de las estructuras anatómicas en pequeño animal con el mismo nivel de detalle proporcionado por los sistemas clínicos.

El objeto de este proyecto consiste en el desarrollo de un conjunto de librerías *software* de bajo nivel para el control de los elementos *hardware* de sistemas de adquisición de imagen micro-TAC de rayos X de alta resolución basados en fuentes de rayos X microfoco y en detectores planos. Las librerías implementadas deben ofrecer una interfaz independiente del *hardware* subyacente y la posibilidad de añadir nuevos dispositivos sin necesidad de modificación en las aplicaciones de alto nivel.

Las librerías diseñadas consisten en un conjunto de módulos individuales para el control de detectores de rayos X, fuentes de rayos X microfoco y controladores de motores, proporcionando una interfaz transparente a la rutina de alto nivel. Las librerías de manejo de *hardware* han sido integradas en una aplicación de alto nivel para la adquisición de datos en sistemas micro-TAC.

La funcionalidad de los componentes individuales fue evaluada en dos sistemas micro-TAC desarrollados previamente. Por su parte, el conjunto completo fue probado en uno de los sistemas, obteniendo un volumen reconstruido que prueba la validez de los componentes software para la obtención de imagen micro-TAC.

Palabras clave: rayos X, TAC, micro-TAC, C++, proyecciones, adquisición.

Index

Image Index

Table Index

Chapter 1

Motivation and Objectives

During the last decades, the numbers of animal models of human disease have largely increased and their use in preclinical translational research has become commonplace. Nowadays it is widely recognized that non-invasive imaging techniques can provide a window into the biology of the living laboratory animals, especially when combining functional and anatomical information. The use of imaging techniques opened up the unique possibility of making repeated measurements of biologic function and specific molecular or genetic events within a single animal over time. In the aforementioned preclinical imaging environments, anatomical information is usually provided by x-ray micro-CT.

For this reason cone-beam (CB) CT systems have been developed using a variety of embodiments and geometrical configurations adapted to high-resolution small-animal imaging (CB micro-CT). The proliferation of these systems is due to the degree of development experienced by the technique, the relatively low-cost of the technology and the ease of integration in combined designs.

The goal of this project is to develop a set of software libraries for the management of hardware components in generic micro-CT systems and to integrate them into a high level application for the acquisition of micro-CT data. With all the presented considerations in mind, the specific objectives of this project are:

- To develop low level C++ libraries for controlling hardware elements in micro-CT systems based on a previously existent set of libraries in C that targeted specific hardware. The libraries must provide a generic interface, allowing the substitution of hardware elements with minor modification to the code and maintaining the same functionality.
- To design and implement a high level application for the acquisition of micro-CT data transparent to the underlying-hardware.
- To integrate the software package in different platforms and operating systems.
- To evaluate the performance of the software.

The work included in this project is framed on one of the lines of research carried out at the Biomedical Imaging and Instrumentation Group at the Departamento de Bioingeiería e Ingeniería Aeroespacial of the Universidad Carlos III de Madrid. A significant part of the developments presented here has been or will be integrated into the preclinical high-resolution scanners manufactured and distributed worldwide by SEDECAL (Madrid, Spain).

Key milestones

This project has a clear task division since the software modules are directly related with the hardware components of the imaging systems. We first developed the libraries for the flat-panel sensor data collection. Then proceed with the mechanical module and the last interface, the x-ray source manager. At the end of every stage, the interfaces were assessed as standalone hardware blocks.

While working on the tube interface, we received the manufacturer specifications and drivers for a new flat-panel sensor working on Windows operating system. Therefore we decide to carry out the necessary modifications to port our interfaces.

Once every block was finished, we integrated them into the acquisition interface and finally assessed the complete software.

The above described work flow is illustrated in the chart below.

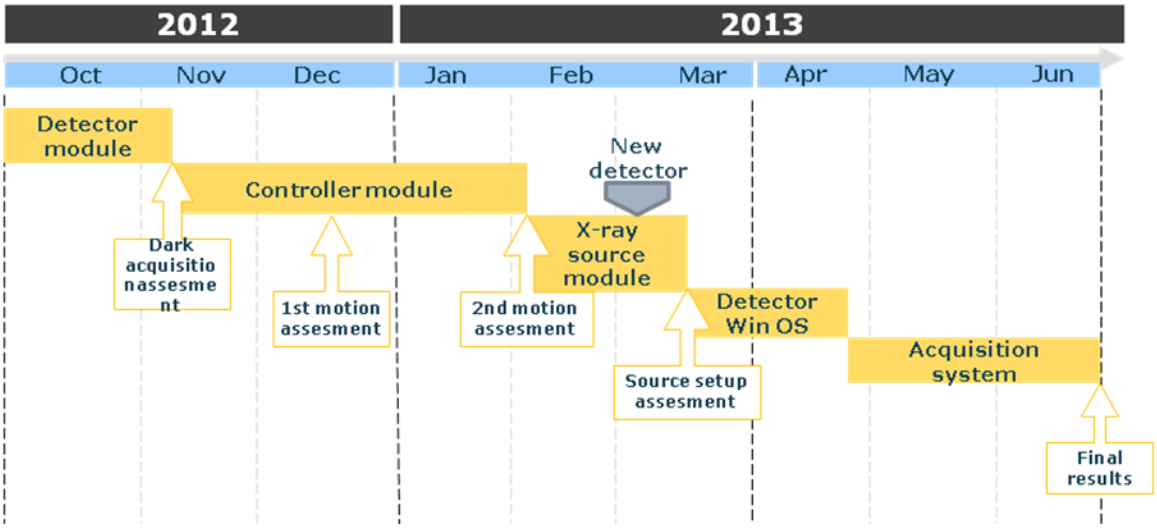


Figure 1. Project workflow.

Target systems

The department is equipped with two CB micro-CT systems and two more on the way, namely:

- FDOT-CT
- Argus PET/CT
- Test bench for Carlos III University
- SuperArgus

Outline of the document

The present document is organized in 6 chapters. Chapter 2 presents the state-of-the-art of the CT technique and an introduction to image formation in x-ray imaging in general and in CT in particular, including common quality degradation sources and image reconstruction. Chapter 2 finishes with a brief introduction to the components of micro-CT systems.

Chapter 3 presents a detailed description of the hardware employed for the project.

In chapter 4 the author presents an overview of the implemented software tools. Following a down-top approach, each module is described in terms of configuration attributes and function prototypes. Finally, the software architecture integrating the low-level modules is presented.

The validation of the performance of the software is presented in chapter 5 that consists on a step-by-step description of the general interface execution and the performed assessments. It includes the middle tests which were performed whenever a low level library was finished, and also the final acquisition program, together with the image reconstruction procedure.

In chapter 6, the budget of the project is presented.

Chapter 2

Principles of CT imaging

2.1 Introduction

Biomedical imaging is the discipline referring to several different techniques that are used to observe a certain property of the body and represent it using visual information in order to diagnose or monitor medical conditions. It has been the breakthrough with highest impact on the clinical and preclinical environments of the last century, becoming essential in daily clinical practice. Indeed, it highly increases the accuracy of diagnostics, the effectiveness of treatment and the chances of abnormality detection.

When using medical imaging techniques, the spatial distribution of a particular property of a biological tissue is measured and represented by the difference in light intensity between the image areas, i.e. by image contrast. According to the source of contrast, two different modalities can be defined: structural imaging, referring to those techniques that provide morphological information on subject anatomy; and functional, which, on the other hand, put up information about functional features such as metabolism or blood perfusion.

X-ray Computed Tomography (CT) can be defined as the structural medical imaging modality that provides a volumetric representation of the attenuation coefficient shown by the different body tissues to the incoming x-ray radiation.

.

This non-invasive imaging technique consists on the acquisition of multiple projection images from different angular positions around the subject. These projection images record the attenuation undergone by the x-ray beam when traversing the subject under study. Thus, the projection images provide line integrals of the physical quantity under evaluation: the x-ray attenuation coefficient.

The acquired projections relate to the distribution of the attenuation coefficient of the different tissues inside the subject by the Radon transform. Radon's theory [1] states the complete characterization of the two-dimensional distribution of a property inside an object by an infinite number of line integrals. According to this, for the formation of the CT image, x-ray attenuation measurements are carried out in all angular directions around the subject, forming a projection dataset for each angular view. Many narrowly spaced data points are measured for each projection for the reconstruction of accurate anatomical images [2] .

Usually the attenuation coefficient of a tissue is referred to that of water to minimize the impact of non-idealities of the acquisition process, yielding the so-called CT number or Hounsfield Unit (HU), which is the most common unit for CT data representation.

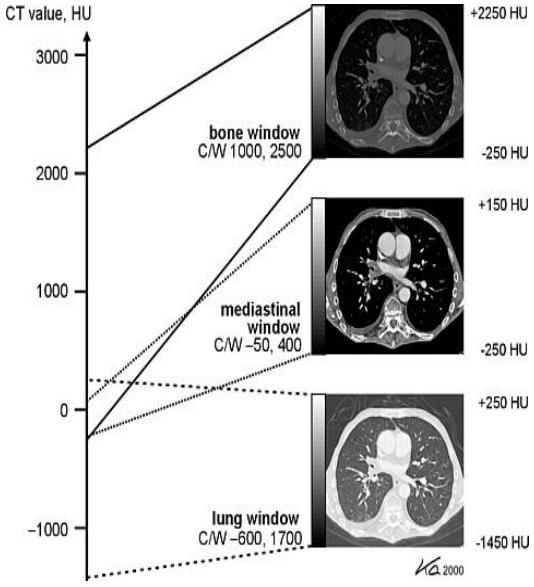


Figure 2. CT number windows [4].

Towards multimodal imaging systems

Nowadays, fields as genomics and molecular biology are on the cutting-edge, regarding both technology and diagnosis. The first term encompasses the study of all the genes of a tissue (at genotype DNA, transcriptome mRNA or proteome levels), sketching the fine-scale genetic mapping of the entire genome's networks [3].The second field, which at some points overlaps with the area of genomics, is focused on the interactions between DNA, RNA and protein biosynthesis [4].

Thus, both disciplines share a common scope: the investigation of the roles and functions of single genes using three-dimensional functional and structural studies, preferably obtaining complementary information.

With modern medical and biological research focused on this common topic, it is latent the urge of non-invasive systems for experiments requiring longitudinal studies over long periods, since disease follow-ups are of high interest. The vast majority of the aforesaid experiments use animals of reduced size, such as rodents, as subjects under study: 95% of laboratory animals are mouse and rat models of human disease, due to the resemblance of their physiology and genetic with that of the human body [5].

This fact has direct implications on instrumentation requirements, namely, the radiation dose shall be carefully reduced and limited to permit the performance of several scans on a single subject, and electronic instruments shall be scaled to retrieve images with a higher resolution while maintaining a low noise level to achieve comparable image quality to that of human body scans.

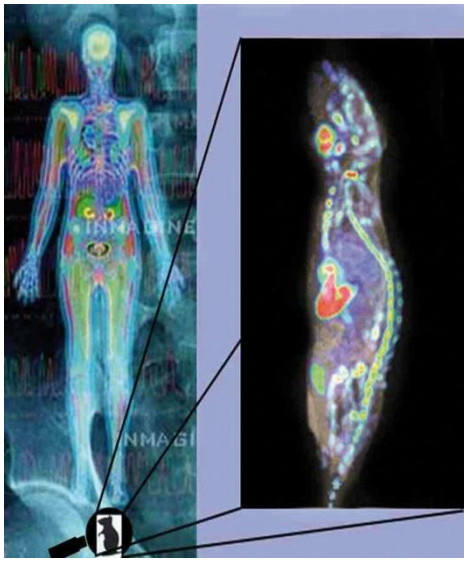


Figure 3. Human-mouse size comparison [5].

In order to suit the resolution and radiation minimization constraints imposed by the size of rodents, new medical imaging instruments shall be designed. At this point CT comes to the rescue offering structural data that, in combination with functional imaging techniques such as Positron Emission Tomography (PET), Single Photon Emission Computed Tomography (SPECT) or Fluorescence Diffuse Optical Tomography (FDOT), provides the accurate anatomic location of functional data in living animals.

From this alliance arises a brand new key interest: increase the compatibility between devices and modalities, i.e. create multimodal imaging systems.

Considerations for small animal imaging

As previously mentioned, there are some remarkable differences between human and small animal scanners which impose critical constraints over instrumentation, namely:

Laboratory animals are a thousand times smaller than humans, requiring a consequent adjustment of image resolution, only achievable by increasing radiation.

Laboratory animals are short-lived, which appeal for minimally invasive in-vivo processes in order to reduce tissue damage during the scan and also to give the possibility of carrying out follow-up studies. Since the animal will be radiated quite often, for this kind of studies dose must be minimized.

As a consequence of the two previous points, a compromise between image quality and radiation dose must be sought.

In addition to the considerations presented above. The animal must remain steady during acquisition. In human imaging, a single apnoea is enough for most of the acquisitions, whereas in the case of animal models, they are anesthetized, creating the need of monitoring of biological signals during acquisition.

The facts exposed above can be translated into numbers characterizing micro-CT scanners and distinguishing them from in-vitro and clinical systems:

	In-Vivo animal CT	***Clinical CT***
Focal spot size	15-200 µm	370-700 µm
X-ray power	10-300 W	60-100 kW
Spatial resolution	50-200 µm	400-1250 µm
Minimum Scan times	18 s	0.33-0.5 s
Detector	Flat panel, CCDs	Solid state detectors: silicon CCDs, UFC, …
Field of measurement	30-100 mm	200-700 mm
Dose	As Low As Reasonably Achievable	

Table 1. Micro-CT vs clinical CT comparison [2, 6-9].

2.2 Fundaments of CT image

Signal Formation

The basic idea of Computed Tomography, as it was previously outlined in the introduction, lies on the following physical principles:

- The x-ray beam goes through the object (body) being attenuated during its travel.
- The total attenuation suffered by the beam is dependent on the different attenuation coefficients of the traversed tissues and the traversed distance across each tissue.
- The information about the accumulated attenuation is carried by the emergent radiation which is recorded forming a projection.

From the last fact it can be deduced that if the beam is emitted from a set of multiple angles, surrounding the body, and forced to cross through the same point for every angle, we will retrieve enough information to determine the punctual attenuation coefficient. Repeating this procedure over all the points of interest we acquire enough information to estimate the spatial distribution of attenuation coefficients within the subject. In other words, the 3D distribution of the x-ray attenuation is reconstructed from either 1D stacked or 2D projection data.

X-ray radiation in medical CT usually is on the range that comprises from 20 keV to 140 keV, where three different radiation-matter interaction mechanisms, whose net effect is the aforesaid attenuation, prevail:

Photoelectric effect: It refers to the interaction between the photon and an electron in the medium in which the photons releases all of its energy and ejects the electron from the atom (figure 4.A). The electron is ejected with a kinetic energy equal to the difference between the energy of the photon and the binding energy of the electron. Outer-shell electrons are the most likely to be knocked. The vacancy left by the electron is almost instantaneously filled up by another electron from a higher energy level and a secondary photon results from the energy difference. The energy carried by secondary photons in biological tissues is usually low enough to ignore them. The photoelectric effect is the dominant effect in high-Z materials and low x-ray energies and it is considered the ideal process for x-ray projection data formation.

Compton scattering: In this type of interaction the impinging photon interacts with an electron from the outer shell of an atom in the medium. The collision frees the electron from the atom whilst the incident x-ray photon is deflected or scattered with partial loss of its initial energy and a different direction as illustrates figure 4.B, that depends on the energy lost during the interaction. Not much information regarding the location of the interaction or

the photon path can be retrieved with such a wide deflection angle and therefore Compton scatter is usually a source of degradation in image quality. Since Compton scattering is an interaction between photons and outer-shell electrons, its probability shows a dependency on the electron density of the material.

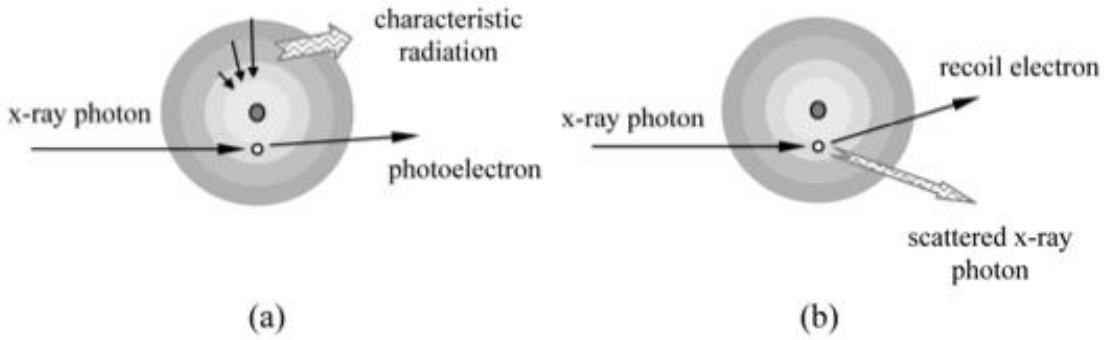


Figure 4. Photoelectric interaction (a) and Compton interaction (b) [10].

Rayleigh scattering: This interaction occurs when electrons of an atom are set in vibration by an incoming x-ray photon, emitting radiation with a wavelength equal to the one of the electromagnetic wave that made them vibrate, i.e. the original photon, and a slightly different direction.

The interactions that produce the highest amount of absorbed or scattered photons are the photoelectric mechanism (for low energy photons) or the Compton scattering (for photons above 50-60 keV, depending on the medium), whereas Rayleigh scattering is noticeable at low photon energies.

The attenuation effect is quantified by the Beer-Lambert law, which describes the exponential relation between the attenuated intensity after crossing the sample and the incident intensity considering a uniform material in terms of density and atomic number and a monochromatic incident x-ray beam.

$$I = I_0 \mathrm{e}^{-uL} \tag{2.1}$$

Where I_0 refers to the non-attenuated x-ray intensity, μ is the unit-length attenuation coefficient of the traversed material and L is the length of material.

Since the slice of a human or animal body is not uniform and presents variable materials with different density values, the overall attenuation characteristics are estimated from the measured line integrals The resulting value corresponds to the projection measurement, p:

$$p = -\ln\left(\frac{I}{I_0}\right) = \int_L \mu(x)dx \tag{2.2}$$

The inherent difficulty lies in the fact that it is impossible to retrieve an infinite number of line integrals, which will unequivocally recover the image. A compromise must be sought between image quality for diagnosis and a sufficient number of lines [10] .

Geometrical configurations for CT acquisition

The geometrical configuration of a CT system describes the arrangement of its hardware components. Provided we will devote section 2.3 to deeply explain the components of a CT scanner, as well as its design criteria, in short, it needs: a x-ray source and a sensor to record the attenuated x-ray intensity. Each and every component is mounted on the mechanical part of the system, allowing translation and rotation movements for highly accurate sampling processes [11].

The evolution of the CT systems has been partly related to the geometrical configurations from its irruption in the XX century to these days.

The first generation of systems turns to punctual x-ray sources that emitted a beam with cylindrical, narrow shape and incident on a single detection element. To acquire enough projections of the area of interest, it combined displacement and rotation movements. The acquisition time for parallel-beam scanners was quite long, increasing the radiation dose and the risk of image quality degradation due to patient motion. Figure 5.A shows the geometrical arrangement of this sort of systems.

The aforementioned issues were partially overcome in the next generation, when equally spaced rays arranged on an enveloping detection arc minimized translation and, therefore, the acquisition time. This configuration was known as fan beam (figure 5.B) and it increased the sample area covered by the source-detector system at each position. However, scanning with this geometry is restricted to a few slices. For even quicker scanning and to minimize detector issues, they resorted to fan-beam with detection ring. Nevertheless, the economic investment required for building the stationary ring did not make this design viable.

Brand new detectors such as Charge Coupled Devices CCDs with flat crystals or flat-panel detectors lead to cone-beam projection, shown in figure 5.C. Samples still focus on a single point with a slight difference: multiple fan-beam planes are collected simultaneously to cover a volume. This current trend answers high-resolution demands for reasonably small active areas, without a strict time constraint [2].

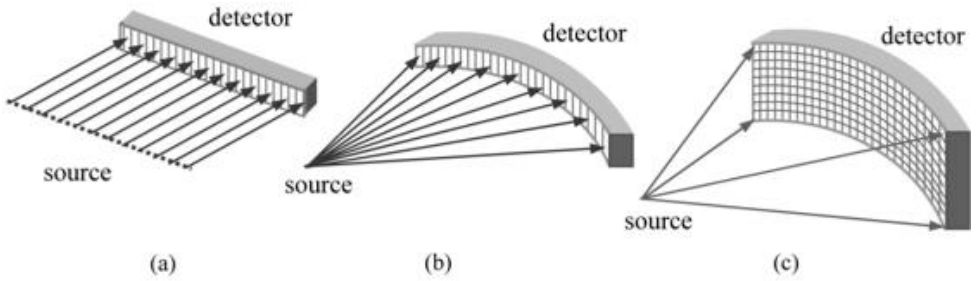


Figure 5. CT geometries: parallel-beam (a), fan-beam (b), and cone-beam (c) [10].

The multi-slice nature of cone-beam CT allows getting the maximum benefit from every acquisition since the detected information can be processed with different methodologies and can be reformatted into sagittal, coronal, oblique or curving planes; or even be displayed as surfaces and volumes.

Up to this point, we have seen how and what does the electronic instrument measure, and with what purpose: reconstructing the structural image. But how do we jump from attenuation values to image?

Image reconstruction

Every point that gathers the emergent x-rays is an element of the virtual matrix on which the computer will work. Each one of the elements is called pixel, when considering 2D, and voxel, when represented in 3D. In other words, from the point of view of the computer, any sample can be considered as a voxelized volume crossed by the x-ray beam. Image reconstruction is the process of obtaining the spatial distribution, $\mu(x,y)$, of the attenuation coefficient through the combination of the projection data from each angular view, $p(t,\theta)$.

The sampling sequence starts with the set of measurements along x-ray paths that are uniformly spaced, forming a projection, as shown in figure 6. Subsequent projections are acquired at different angles, maintaining a constant increment between adjacent views, until the 360 degrees are covered.

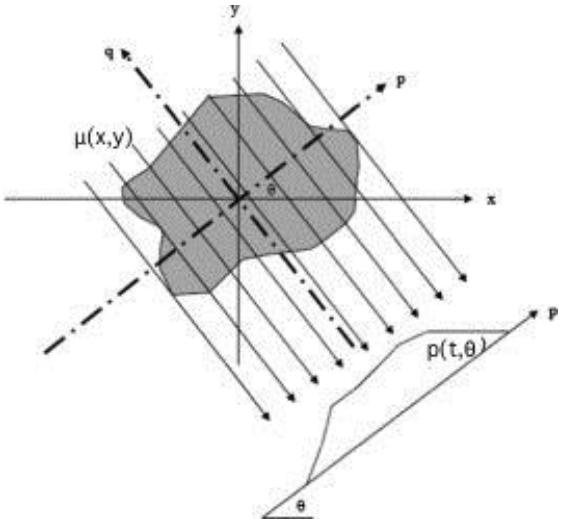


Figure 6. Schematic illustration of the acquisition of a projection dataset [12]

Once the sampling data is available, reconstruction of the attenuation function can be estimated through different approaches. The most intuitive and extended approach is the inverse Radon transform. The image to reconstruct can be thought of as a cross section through the specimen, in which intensity values represent the attenuation coefficient. Each attenuation measurement, i.e. its projection value along its ray *(t,θ)*, gives a value of the integrated attenuation coefficient along the object, which corresponds to the Radon transform through the object. The inverse Radon transform provides an approximation of the image but significantly blurred.

The source of the blurring presented by the backprojected image is the overestimation of low frequencies and underestimation of high frequencies when adding data from all projections. Adding a frequency-dependent weight to the projection data can compensate this difference in sampling density [10]. Note in figure 7 the overlapping of the different projections in the low frequency area and how weighting can compensate this effect.

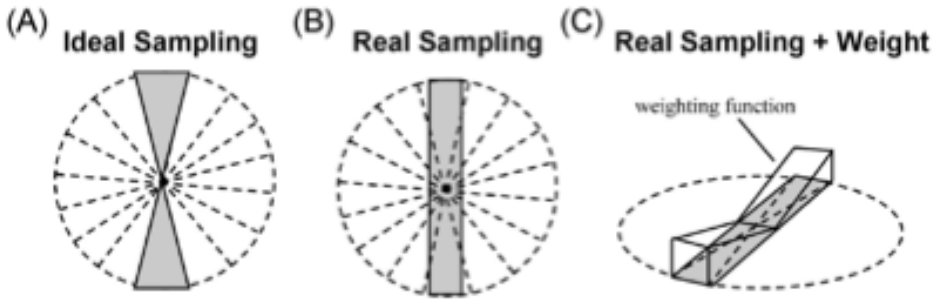


Figure 7. Sampling patterns of the frequency space: ideal situation (A), real situation for backprojection approach (B) and the compensated situation (C) [10]

Analytical algorithms such as the Filtered Back Projection (FBP) follow this principle. In general terms, this algorithm is widely used because of its low sensitivity to space domain interpolation error and its reconstruction speed. It is based on the concept that each slice is obtained by convolving the image with a ramp filter, presenting some limitations due to its finite projection bandwidth and finite projection order [10].

Alternatively, the iterative approaches assign specific attenuation values for each element, without a priori knowledge of the subject itself and initially assuming homogeneity. Attenuation parameters are updated depending on whether the calculated projection is overestimated or underestimated, requiring several iterations before converging. One example of this kind of methods is the Algebraic Reconstruction Technique (ART) [13]. However, despite the computational load being higher than in analytical approaches because of the need of performing several forward and backward projection operations, the use of GPUs and their massive parallel computing capabilities has notably speed up iterative algorithms [14].

Limitations of the CT technique

The accuracy of projection measurements is threatened by the difference between ideal conditions and those of real scenarios. Some of the assumptions or approximations may no longer hold in real imaging conditions. Such non-idealities impact diagnosis performance because of the distortions they introduce in the image, which in some cases, mime pathologies.

Some sources of errors are scattered radiation, that mainly adds a low-frequency bias that increases with radiated volume to the attenuation measurements; non-linearities of detector due to dark current biasing, hysteresis or gain inequalities; and the polyenergetic nature of the x-ray beam, reinforcing the energy dependency of the attenuation factor, as shown in the equation below [10] .

$$I = I_0 \cdot \int_0^E p(\varepsilon) \exp(-\int_0^\infty \mu(\varepsilon, x) dx) d\varepsilon \tag{2.3}$$

Where $p(\varepsilon)$ represents the incident x-ray spectrum, I_0 and I represent the incident and transmitted x-ray intensities with energy E and $\mu(\varepsilon,x)$ is the linear attenuation coefficient of the object at the same energy.

This energy dependence is usually ignored by later image processing and reconstruction stages, to a large extent due to the lack of data to incorporate it in the processing chain, giving way to the generation of quality degradation effects such as beam hardening. It is desirable that the beam approaches a monochromatic, i.e. mono-energetic, beam for

minimizing the number of issues affecting image generation. Other advisable characteristics of the x-ray source are the followings [11]:

- It must be able to produce enough rays within a short period of time.
- The user shall have the chance to choose the energy of the rays beforehand.
- Rays should be reproducible.
- Beam coherence and radiation losses have to suit security standards.

We have already outlined some compromises and trade-offs that shall be sought in compliance with image quality for diagnosis. Pushing too far any design parameter will lead to image quality degradation. Moreover, the reconstruction technique assumes the consistency of all acquired values, reflecting every acquisition error in the reconstructed image. This systematic discrepancy between real attenuation and the computed CT value is known as CT artifact and can be recognized in the reconstructed image in which undesirable lines, shadows, rings or other effects appear.

The sources of artifacts can be classified in four categories: artifact due to physics (beam hardening, lack of photons, subsampling, aliasing), due to subject (patient motion, metallic implants, truncation), due to electronics and mechanics (poor calibration and limitations of the system) and due to multi-slice scanning (data inconsistencies may appear in CBCT because Tuy-Smith condition is not hold [15]). There are two strategies to combat artifacts, through correction or through avoidance [5]. In figure 8 we can visually assessed artifacts belonging to the different categories defined above.

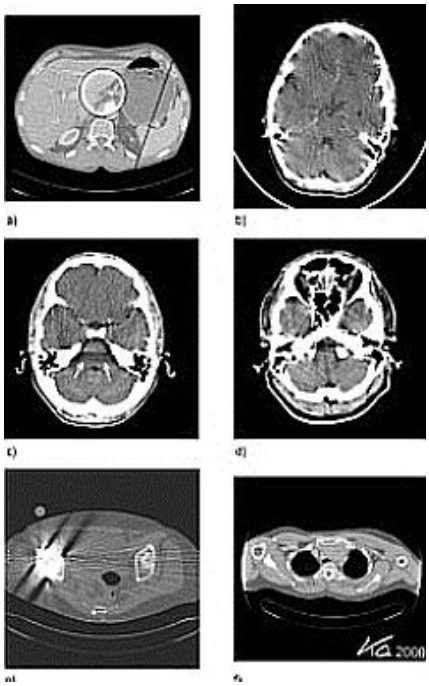


Figure 8. Artifacts due to failure of electronics a), patient motion b), beam hardening c), partial volume effects d), metallic implants e), or patient exceeding the field of measurement f) [2].

Image pre-processing: gain and dark maps

Raw image data provided by the FP detector is contaminated by structured noise that shows as deterministic patterns arising from the structure of the imaging sensor. This correlated noise has to be removed to enhance the quality of the projection data and avoid artifacts in the reconstructed volume. In particular the raw signal is contaminated by two types of correlated noise, namely dark signal and gain heterogeneity of the detector [16].

The dark signal is the image that results only from random generation of electrons and holes in the reverse bias photo-detectors in the sensor area that generates a small current even when no radiation is reaching the sensor [17]. Since this signal is always positive, it has a non-zero mean with a random component superimposed. This non-zero average signal forms an offset image that depends on the image integration time and temperature of the sensor and that must be subtracted from the acquired data. The dependence of the offset on temperature makes it advisable to acquire the correction data just prior to the micro-CT data acquisition.

The second component of correlated noise arises from the small differences in gain of the different individual pixels that generate slightly different signal values for a uniform flux of radiation. The gain map of the sensor is usually estimated by averaging a set of flood images for which the detector is irradiated with the beam set at the acquisition protocol configuration in terms of kVp and filtration and for a value of exposure sufficiently separated from the saturation point of any pixel in the detector.

The straightforward way to correct the acquired data for dark and gain is the so-called flat-field correction [18, 19] which consists in the subtraction of the dark offset and the subsequent division by the dark-corrected gain map, according to the following expression [20], for a pixel with coordinates (u, v):

$$C(u,v) = \frac{R(u,v) - \overline{O}_R(u,v)}{\overline{G}(u,v) - \overline{O}_G(u,v)} \tag{2.4}$$

Where R is the acquired data, O_R is the offset data acquired for the temperature conditions of the acquired data, G is the gain map and O_G is the offset image for the temperature conditions of the gain map.

While the flat-field correction approach provides sufficient image quality for most applications, it inherently assumes a perfect linear gain curve for each pixel. In real detectors, however, it remains some degree of non-linearity in the gain curve of the pixels [21], causing the equalization of the pixel response to leave slight differences between the response of adjacent pixels that in turn translate into artifacts in the reconstructed image. This effect is more evident for signal levels far from the calibration point.

2.3 Micro-CT components

Scanner configuration

An overall view of a fDOT micro-CT system with shielding elements in place but without outer casing is shown in figure 9. It is used as reference model for any micro-CT system; therefore, fDOT specific components are labeled but not described. Figure 10 shows a schematic depiction of the connections and communication channels between the different parts of the scanner.

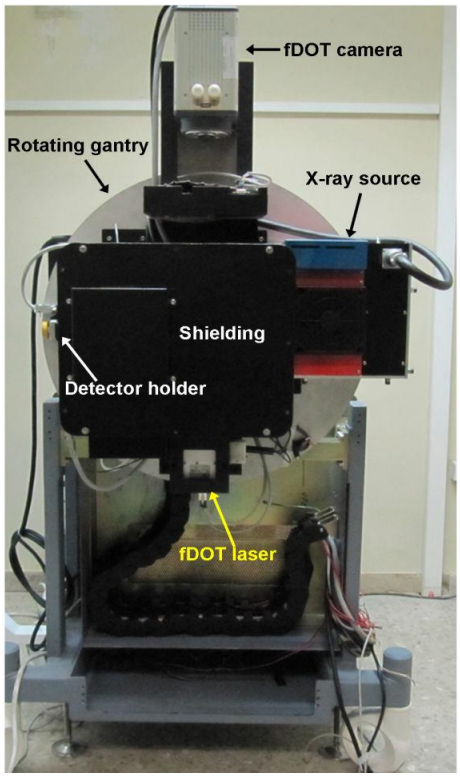


Figure 9. Overall view of the fDOT micro-CT system with the shielding elements in place [22].

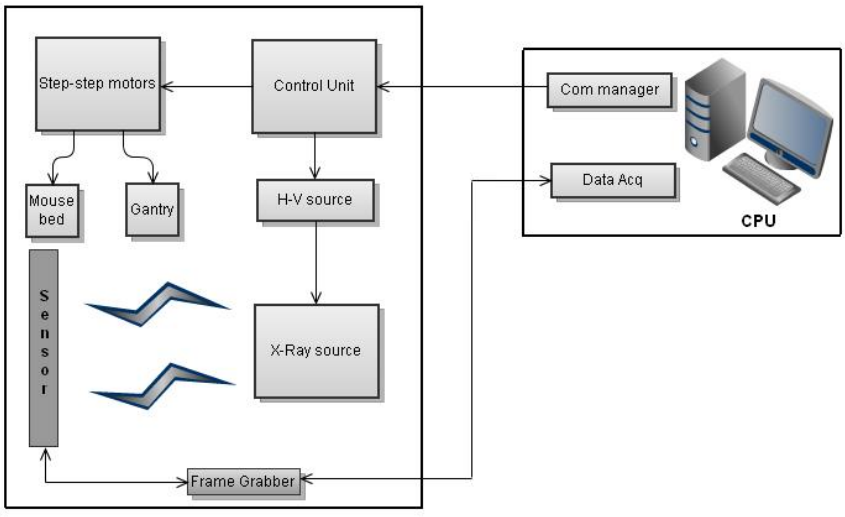


Figure 10. Block diagram of a micro-CT system.

Gantry

This term refers to the mechanical part of the system. Specifically, it encloses the ring and the chassis.

X-ray Source

This is the element in charge of generating the x-ray radiation. The x-rays are created by the collision of electrons, which have been accelerated using high voltages, with a metal target (tungsten for the case of study), the anode. It is this transference of energy, whether partial or total, what generates the x-rays. There are two main situations prone to generate radiation, namely:

- High-speed electrons are forced to decelerate by the electric field of the target nuclei, generating what is called *Bremsstrahlung* radiation. In the worst-case scenario, high-speed electrons hit the nucleus directly. Although it is the most unlikely mechanism, it is also the one which generates higher amount of energy since the entire kinetic energy is converted to x-ray energy.
- High-speed electrons interact with target electrons with enough energy to eject them from their electronic shell. When outer-shell electrons fill in the vacancy, characteristic x-rays are emitted.

Diagnostic x-rays are on the range of mid-soft x-rays, due to their moderate penetrating capability. On the other hand, they provide remarkable high amount of low-contrast information. This is the reason why x-rays with wavelengths ranging between 0.1nm and 0.01 nm (energy range 12.4-124 keV) are useful for diagnosis.

As abovementioned, this scanner is meant for small animal imaging. Therefore, to not degrade image resolution, it shall make use of an x-ray tube whose focal spot diameter remains as reduced as possible, i.e. microfocus x-ray sources are the most appropriate.

Filters of different materials are incorporated into the beam to improve beam mono-chromaticity and reduce both radiation exposure and image artifacts.

Stability of the x-ray source is also highly appreciated to maintain the noise and signal characteristics of the projection data across different angular views.

High-Voltage Power Supply

This component fixes the x-ray source working settings, in terms of acceleration voltage and anode current. The current supplied shall keep the x-ray sensor from saturation whilst getting good sensing performance at maximum acquisition speed. The combination of these two parameters shall never exceed the power limit imposed by the x-ray source constraints.

Thanks to an interface between the working station and the control unit, the user can set the working point in the defined ranges.

X-ray detector

Detectors are meant to be fast and highly sensitive and, at the same time, stable. If these characteristics are achieved, the number of required correction datasets is drastically reduced and so does the acquisition time. Detectors in small-animal micro-CT are integrated in the gantry, whose motion demand light components. For this reason, flat-panel digital detectors are substituting CCDs connected optically to a scintillation screen in despite of their wide spread use. The so-called flat-panels are semiconductor-based light detector matrices, which compete in resolution and image quality with CCDs but with a more compact and lighter design. Their weakness is the speed, being relatively slower.

From the point of view of detection, we can find two different approaches: direct or indirect conversion. In any case, the shared fundament is that flat-panel devices convert x-ray photons, i.e. primary quanta, to electric charges that are gathered and analogue-to-digital converted [23] .

We talk about indirect conversion if the imager has an x-ray phosphor or scintillator screen to convert absorbed energy into visible light. The photodiode layer on the surface of the array produces photo-induced charge within each detector element and the aforementioned charge is stored in a local capacitor.

Opposed to indirect detectors, direct flat-panels use a semi-conductor material layered between two electrodes and electron/hole pairs are directly produced as a result of x-ray energy absorption. A high-voltage bias placed between electrodes separate the charge pairs with negligible side spread.

While both procedures allow high intrinsic spatial resolution, indirect conversion is preferred for small animal [21].

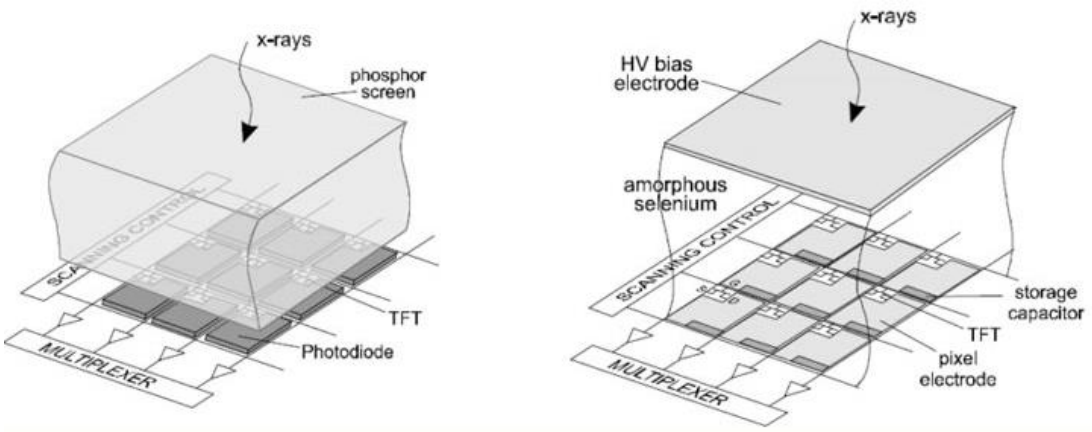


Figure 11. Direct (left) vs. indirect (right) flat-panel detector [21].

Frame Grabber

The device in charge of generating the synchronization pulse that starts up the pixel-acquisition line, and afterwards, of transmitting what the sensor reads to the hard disc allocated in the working station.

The user can define the exposition time or the photodiodes integration interval, acting directly over Signal-to-Noise Ratio and deciding the acquisition protocol [11].

Working Station

A master CPU hosts the calibration, register and control software, as well as the user interface. It is also where image reconstruction from the acquired data takes place.

2.4 Market research study

The increasing demand of this sort of imaging tools has contributed to settle micro-CT systems as part of the health instrumentation market. We present in table 2 a comparison among the best-rated models, which yields a good picture of the state-of-the-art of the hardware.

Device **Company**	**SuperArgus** **SEDECAL**	**eXplore Locus RS** **GE Healthcare**	**Inveon** **Siemens**	**Xtream CT** **Scanco**	**SkyScan 1176** **Bruker**	**TomoScopeSynergy** **CT Imaging**
						
Specifications						
Resolution (μm)	20, 50, 100	45, 90 (10%MTF)	20, 40	104 (10%MTF)	15 (10%MTF)	80 (10% MTF)
FOV diameterxlength (cm)	12x45	4.5x8.0	8.4x5.5 , 10x10	12.6x24	6.8 x 20	6.5x15
X-rayenergy (kV)	20-110	35-80	35-80	60	20-90	20-65
Focal spot (μm)	15-80	10	6, 50	80	?	50
Dector panel	CMOS	CCD	FP 125-165 mm	FP 170 mm	CCD	FP
Min acq time (s)	15	15	0.15	80	<60	15
Max.power (W)	50	40	40	60	25	Mid. power
Radiation safety (υSv/h)	<1 at any point on the instrument surface during scanning					
	[7]	[8]	[9]	[6]	[24]	[25]

Table 2. Comparison of current micro-CT system models

Chapter 3

Hardware

The x-ray system is a simple PC-based setup, with an x-ray tube and digital detector controlled by data acquisition cards in the PC. To generate CT data, the object to be scanned is either placed on a steady stage around which a mechanical system, with detector and x-ray source mounted, rotates, or on a rotating stage between the x-ray tube and the detector. Three out of the four target micro-CT systems are built upon this first named design. The test bench design for Carlos III University implements an in-vitro configuration where the sample, instead of the x-ray tube-detector assembly, rotates.

Such system designs have maximum flexibility to allow to easily vary both geometry and acquisition parameters, in order to suit user's demands. There are small differences among models of components. Hardware specifications are described in the current section.

3.1 Overview of the equipment used in this project

The different target systems were listed at the beginning of the manuscript. However, a more detailed description of the specific models and their components is included below.

- FDOT-CT is built upon a Hamamatsu flat-panel detector, Technosoft controllers for the mechanical system and a Hamamatsu x-ray source.
- Argus PET/CT system also uses a Hamamatsu sensor but the motion controllers are manufactured by API Controls and the x-ray source by Oxford Instruments.
- Test bench for Carlos III University has a Dexela flat-panel detector, Technosoft motion system and a Hamamatsu x-ray source.
- SuperArgus system relays on a Dexela flat-panel detector, Technosoft motion system and a Hamamatsu x-ray source.

These model characteristics are graphically summarized in figure 12.

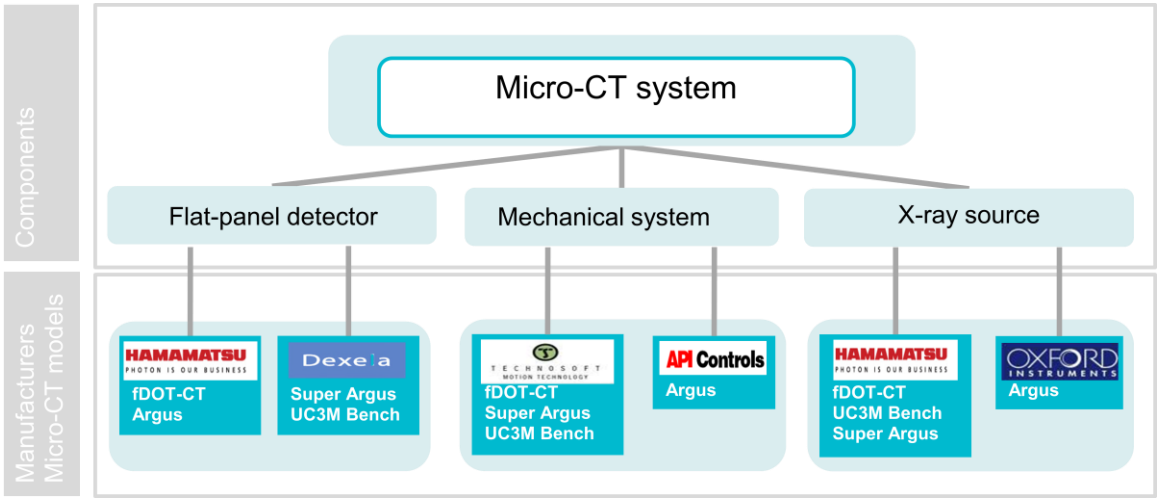


Figure 12. Graph relating the components of CT-systems with the target systems of this project and the hardware manufacturers.

3.2 Flat-panel detectors

Two different flat-panel detectors were contemplated for the development of the controlling libraries. A description of their main features and a brief explanation of its main parts are provided in the following subsections.

Hamamatsu detector

One of the x-ray flat-panel detectors used is the model C7940DK-02 (figure 13.A) (Hamamatsu Photonics K.K., Hamamatsu, Japan). A detailed description and the evaluation of some of the performance parameters can be found in [26, 27]. It is an indirect flat-panel

detector based on a CsI:Tl scintillator screen, as it can be seen in figure 13.B, and a high-fill-factor CMOS image sensor.

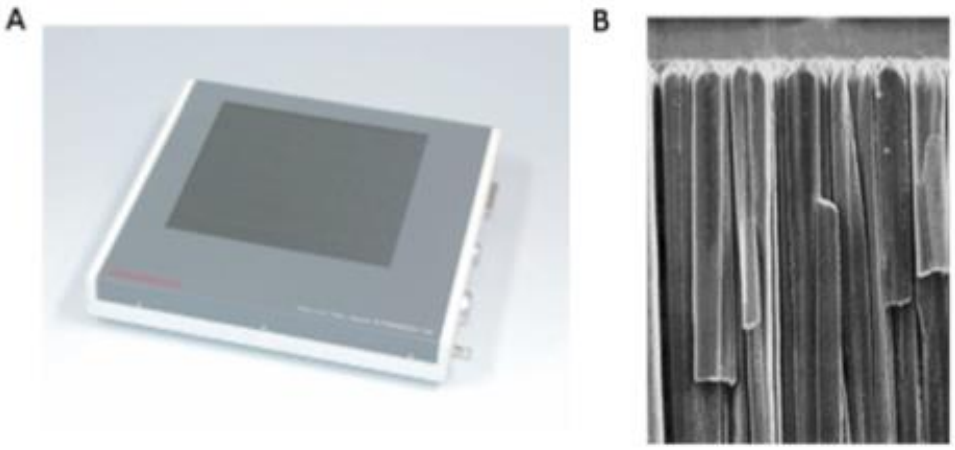


Figure 13. Hamamatsu C7940DK-02 X-ray flat-panel detector (A) and detail of the needle-like CsI:Tl crystal structures on the scintillator layer (B). Images courtesy of Hamamatsu Photonics K.K.

The x-ray photons are stopped on a 67.6 mg/cm^2 (0.15 mm-thick) microcolumnar CsI:Tl layer, directly deposited over the CMOS photodiode matrix surface (figure 13.A). The active pixel elements of such matrix are connected to CMOS transistor switches that enable detection and read-out, and offer a high fill-factor (79%).

The CMOS sensor makes use of on-chip signal amplification channels with low noise level, and an offset suppression circuit based on correlated double sampling (CDS), assigned to each of these channels. This design achieves a high degree of image uniformity and a low noise level. However, correction tasks must be performed after the acquisition of an image in order to obtain the best achievable quality, getting rid of residual non-idealities.

The features of the flat-panel detector as reported by the manufacturer are summarized in the table 3.

Parameter	***Value***
Pixel size	0.05, 0.1, 0.2 mm (Binning 1x1, 2x2, 4x4)
Photodiode area	120 x 120 mm
Number of pixels (Total/Active)	2400 x 2400 / 2240 x 2344
Frame rate	2, 4, 9 fr/sec (Binning 1x1, 2x2, 4x4)
Noise (rms)	1100 electrons
Saturation charge	2.2 M electrons
Dynamic range	66 dB
Resolution (MTF 0.2)	5.8 lpmm

Table 3. X-ray flat panel detector characteristics.

Dexela detector

The second x-ray flat-panel detector contemplated for the software design is the 2315 model (figure 14, Dexela Limited, London, UK). A complete description and an evaluation of some of the performance parameters can be found in [28]. It is an indirect flat-panel detector based on a 150 µm structured CsI:Tl scintillator screen and a high-fill-factor CMOS image sensor to detect the optical photons.

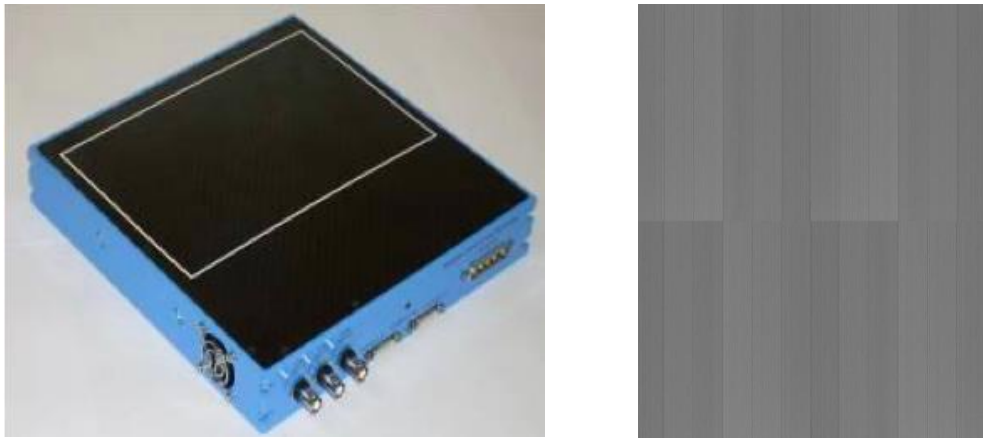


Figure 14. Dexela X-ray flat-panel detector (left) and dark image sample (right). Dexela Lmt [29]

The x-ray photons are stopped in a 67.6 mg/cm2 (0.15 mm-thick) CsI:Tl scintillator and the resulting visible light is then detected by a 3072 x 1944 CMOS active pixel matrix, with a pixel size of 75 µm. Full well capacity is the term used to defines the amount of charge an individual pixel can hold before saturating. Each pixel contains an option for switching between two separate levels, high full well (HFW) and low full well (LFW) modes. The typical fill factor of this detector is around 84 %. The detector offers the possibility of extending the area of a pixel using pixel binning (1x1, 1x2, 1x4, 2x1, 2x2, 2x4 4x1, 4x2 and 4x4 pixels).

This design reduces the image integration time and offers two different working modes, one for high sensitivity protocols, where the x-ray flux is minimal, and one for high-dynamic range protocols.

The main features of the detector are shown in table 4.

Parameter	*Value*
Pixel size	0.075, 0.15, 0.3 mm (Binning 1x1, 2x2, 4x4)
Photodiode area	145 x 115 mm
Number of pixels	1944 x 1536/ 1934 x 1536
Frame rate	26 - 86 fr/sec (Binning 1x1 - 4x4)
Noise (rms)	309 electrons (HFW) 126 electrons (LFW)
Saturation charge	1.4 M electrons (HFW) 0.4 M electrons (LFW)
Dynamic range	73.1 dB (HFW) 70.1 dB (LFW)
Resolution (MTF 0.2)	6.3 lpmm

Table 4. Dexela x-ray flat panel detector characteristics.

3.3 X-ray source

The micro-CT system makes use of a tungsten anode microfocus x-ray source (L9631-MOD2, Hamamatsu Photonics K.K., Hamamatsu, Japan). The source has 40 to 110 kVp energy range and maximum anode current of 1 mA, limited as a function of kVp to deliver a maximum power of 50 W. The focal spot size of the x-ray beam varies linearly between 15 and 80 μm with power, for power values larger than 6 W.

A fixed collimator made of a tungsten-bismuth alloy shapes the x-ray beam to cover the detector area and a filter holder is included to place filter materials for the x-ray beam [25].

This applies for all target system except for the Argus system, which uses a microfocus x-ray tube with a tungsten anode, the Apogee XTG5011 (Oxford Instruments, Oxforshire, UK). In this case, the nominal focal spot size is 35 μm.

3.4 Motion system

The detector and x-ray source are allocated on a rotating gantry. This gantry consists of an 80 cm-diameter aluminium plate that holds both components and a rotating motion stage model RV350PP (Newport Corp., Irvine, CA).

The sample is placed inside the FOV of the system by means of a linear motion stage, equipped with a stepper motor. There is also a rotation stage that is either built upon a mechanical ring or the system holding the object to be scanned.

The motorized stages included in the system are managed by servo drives with integrated motion controller boards, model ISCM 4805 (Technosoft S.A., Neuchatel, Switzerland) or DM-224i (API Controls Inc, NY) [30, 31]. The individual drives communicate via a CAN bus which end side is connected to the acquisition computer by a RS-232 serial link.

3.5 Working station

All the components included in the systems were connected to a regular personal computer in charge of the control of the whole system.

In the case of Argus and fDOT-CT models, the acquisition computer ran a Linux operating system (Fedora Core 9). Whereas a Windows operating system (Windows 7) was required for the Super Argus and UC3M Bench scanners.

The abovementioned personal computer was configured to host both operating systems, so that the developed libraries could be tested and their compatibility validated.

Chapter 4

Software interface design

This chapter describes the architecture of the developed software: how the modules are divided into classes and how they are integrated in an interface for the acquisition of CT data.

The development presented here was based on two sets of existent software designed for two of the target systems, the Argus CT and the fDOT-CT. Each one of these two systems ran dedicated software, specifically targeted to the particular hardware of the system. The existent software tools were developed in C language.

As was previously mentioned, the software developments included in this project had the main goal of providing a set of software libraries, offering an interface independent of the underlying hardware, allowing the possibility of modifying hardware components without any change in the upper level acquisition routines. For the last reason, we have designed a set of C++ libraries (one for each of the main hardware components), with a two layer architecture. Each library consists of a set of C++ classes managing the low level hardware and an intermediate level layer consisting on a single C++ class interfacing with the high level acquisition routines.

The low level objects are the units on which the system interface will relay since they are the ones directly interacting with manufacturer's hardware libraries. There is an evident analogy

between hardware and software modules. Above them, three classes were design for each one of the main subsystems of a micro-CT system: detector, gantry and x-ray source.

4.1 Software architecture

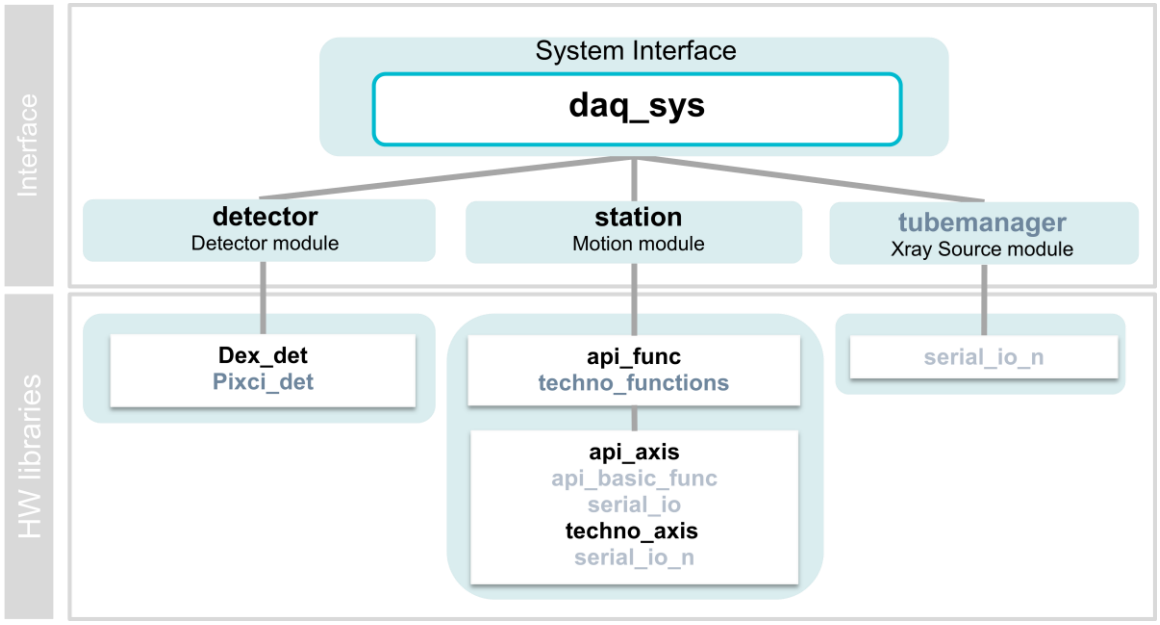


Figure 15. Interface architecture.

The diagram in figure 15 shows a schematic depiction of the software architecture proposed in the present project. The upper box on the diagram (*daq_sys*) in figure 15 shows the high-level application for the acquisition of CT data, making use of the individual hardware libraries.

The developed libraries integrated into the system acquisition interface (*daq_sys*) are:

- the detector and frame grabber module (*detector*),
- the motion module (*station*),
- and the x-ray source interface (*tubemanager*).

In order to provide a transparent interface to the high-level applications, regardless of the underlying hardware, additional classes were required as can be seen in the lowest box (see figure 15). *Dex_det*, *api_func*, *api_axis* and *techno_axis* were developed entirely as part of this project. *Pixci_det* and *techno_functions* were part of an older version of the software and needed to be converted into C++ classes and modified to meet the hardware requirements. And the C libraries *api_basic_func*, *serial_io* and *serial_io_n* needed no modifications. In figure 15 this classification is represented as coloured names: black for the new developments, light-gray for existent, but modified, developments and dark-gray for native code.

4.2 Detector module

The sensor makes use of a frame grabber card to interface the detector from the controlling computer. A schematic depiction of the connections between detector, frame grabber and computer is shown in figure 16.

Both detectors contemplated in the presented development connect to an Epix frame grabber card (Epix Inc., Buffalo Groove, IL) and make use of the functions of XCLIB library [32]. When first accessing the frame grabber, it has to be configured according to detector specifications. For this purpose, when the system is initialized, a configuration file created with the XCap software (Epix Inc., Buffalo Groove, IL) has to be loaded.

The library has been designed to work in Windows and Linux environments. However, the Dexela detector is accessible only in Windows due to restrictions of the manufacturer.

To achieve this duality we implemented a function that retrieves a handler to the appropriate library as a function of the operative system (OS) in the target computer, using the dynamic library handlers provided by the particular OS.

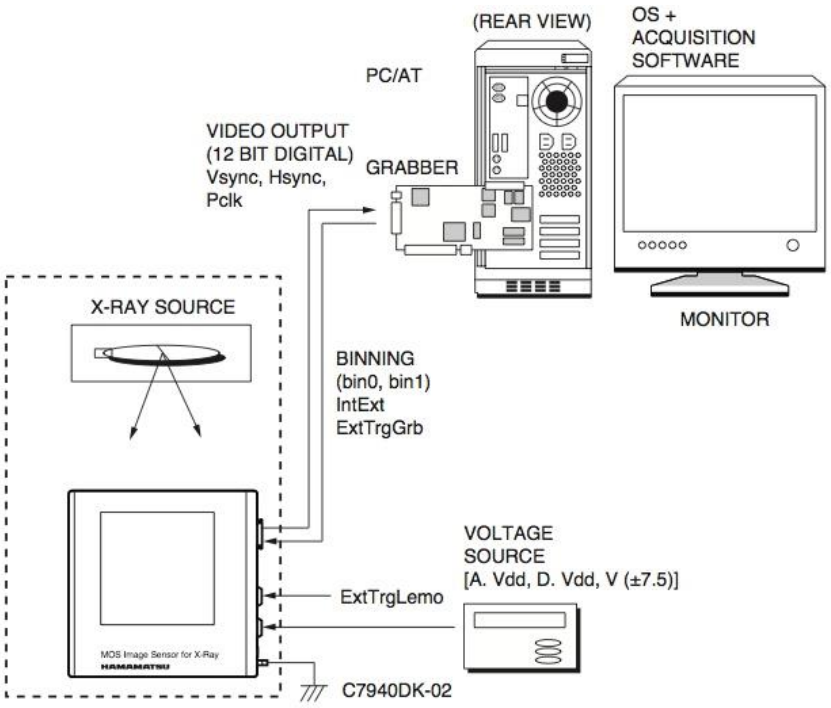


Figure 16. Wiring and communication diagram of the flat panel sensor and frame grabber [18].

Regardless of the flat panel model, the C++ class –either *pixci_det* or *dex_det*–, which imports the XCLIB library, must contain the following parameters:

- Detector coordinates, which include the upper and lower limits of the active area.
- User-selected averaging block size.
- Path to the format file.
- Flag for external triggering.
- Variable storing the image exposure time.

The classes interfacing implement functions of three different categories: detector initialization and access to parameters, control tasks and acquisition routines. The set of functions is explained in the sections below.

Detector initialization and access to parameters

The detector object can be initialized using a set of default parameters or using the values specified in the system configuration file. In the event of using the empty constructor, the default parameters for the configuration of the detector are extracted from predefined configuration files.

Regarding the detector area, the upper horizontal and vertical coordinates are initialized to zero and the lower horizontal and vertical coordinates to -1 (meaning to use the complete detection area). Frame grabber status flags are assigned to zero, indicating that the detector is not open, it is idle, and that the continuous acquisition mode is not activated.

If the user has specified the detector area to use, the number of images to acquire and the format file beforehand, then the constructor of the detector object will take these values into account and construct the corresponding object with user-defined values.

With the aim of retrieving or modifying the parameters after the initialization, get/set functions were implemented for every parameter of the detector managing class, so that they can be easily accessed within the high-level application.

The library also provides methods for querying for parameters for parameters that are dynamically updated during the image acquisition, such as the captured field counter or the buffer counter. This information is of paramount importance for synchronization tasks during data acquisition.

Table 5 provides a reference of the functions we have just mentioned.

Function	***Description***
do_vidsize	Query about image resolution: dimensions, colors and bits per pixel.
do_imsize	Query about image frame buffers: size, number of buffers, number of boards.
set_UD_ExpTime	Sets exposure time
get_tam_tot_x	Query about the number of pixels in x coordinate.
set_tam_tot_x	Sets the number of pixels in x coordinate.
get_tam_tot_y	Query about the number of pixels in y coordinate.
set_tam_tot_y	Sets the number of pixels in y coordinate.

get_ave_fr	Query about the size of the averaging block.
set_ave_fr	Sets the size of the averaging block.
get_trigger	Query about triggering mode.
set_trigger	Sets triggering mode.

Table 5. Functions to access parameters.

The library can be extended by including additional functions which are only available for a particular hardware. These functions are only accessible if the appropriate hardware is present. One example of this kind of functions is the setting of the waiting time between the acquisitions of two consecutive frames in a sequence that is available only for the Dexela detector.

Once the attributes of the class are initialized, the interface is ready to interact with the hardware device and perform data acquisition routines.

Control functions

By opening the imaging board after the system initialization but before programming any data acquisition, the connection to the frame grabber card must be started. Once the required tasks are finished, this connection has to be closed.

In order to perform these opening and closing of the imaging board, two initialization functions have been developed: *do_open* and *do_close*. The abovementioned format file is specified at the time of the detector object creation and accessed within *do_open*. For the flat-panel detector control our library provides the functions described in table 6.

Function	***Description***
do_open	Opens the imaging board communication channel.
do_close	Closes the imaging board communication channel.
do_pingpong	Initiates continuous capture into buffer 1 and buffer 2, alternately.
do_unlive	Stops the acquisition after the current field.

Table 6. Detector control functions

Acquisition routines

Three different approaches for image acquisition were implemented, namely single frame acquisition –snapshot–, multiple frame acquisition in sequence or multiple frame acquisition in continuous mode:

- **Single frame**: Acquiring a snapshot is a simple process that calls *do_snap*. The process actively waits for the last field acquired to change its value. This means that a frame has been acquired and temporally saved in the block memory indicated by current buffer field. The method concludes by dumping the image into a user buffer previously allocated in the high-level application.

- **Multiple frames**: In general, a single image does not provide enough data for imaging purposes. The user requires larger number of images at different angular and/or linear positions. Launching free-run video capture takes advantage of the maximum data rate provided by the detector [18], thus reducing acquisition time and delivered dose. These processes initiate a continuous capture of images into a couple of buffers from the grabber, alternately, i.e. ping-pong style. In multiple-frame modes, the value of the averaging-block–size variable is the number of images to be averaged in each angular position. Both sequential and continuous acquisitions serve this purpose:

    - **Frame sequence**: This method captures a sequence of *n* images with *m* averaged frames for each one of them. The total number of desired images, the size of the averaging block and the pointer to user allocated buffer are the input parameters of the function. The frame grabber must be already running in video capture. Every image is stored one after the other in user buffer, until the number of acquired images equals the desired number of images. The capture won't be stopped inside this method, although no other image will be dumped from the board's buffers.

    - **Continuous acquisition**: A call to this mode initiates continuous capture of images with *m* averaged frames in each one of them. Images are stored in user-dedicated memory. The user is responsible of memory allocation for the storage. Unless a second process toggles the end-of-acquisition flag, images will be acquired non-stop. As it can be inferred, two processes running simultaneously are needed for this scheme, as it will be explained in detail in section 5.1. As the capture was started by this function, it must also be stopped in this scope, terminating after the current field or frame.

To sum up, the acquisition functions are shown in table 7.

Function	*Description*
ACQ_snap	Captures a single image.
ACQ_sequence	Provides a sequence of n images averaged in blocks of m-frames.
ACQ_cont	Initiates continuous capture, averaging in blocks of m-frames each image.

Table 7. Acquisition mode functions.

4.3 Motion module

The motion library is formed by a set of C++ classes divided in two levels. The upper level (*station*) provides a hardware-independent interface, while the lower level classes provide access to the hardware, adapting the functionality offered by *station* to the underlying hardware. Figure 17 illustrates the structure of the motion libraries.

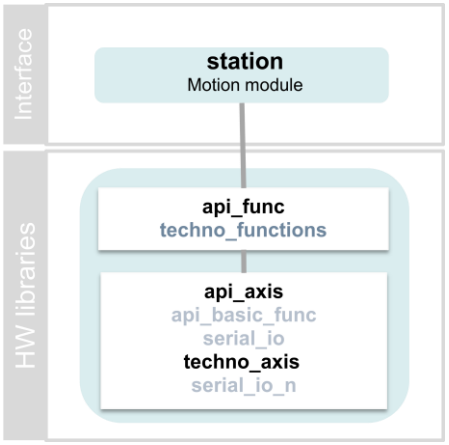


Figure 17. Detail of figure 15 isolating the motion module structure.

Depending on the manufacturer, properties of the motor controllers are managed through the corresponding low-level class, *techno_axis* or *api_axis*. With respect to the kind of axis, the functions perform differently depending on whether the particular instance to an object of the low-level class refers to the bed, the gantry or the detector. The reason for this is that their movements required specific unit conversions and limits. The ring rotates whilst the bed and the detector linearly displace, for example; their speed and distance are independently selected, etc.

Therefore the object storing the parameters of the axis contains the basic information about the hardware it represents, as listed in table 8.

Technosoft	*API*
Identification integer	Identification integer
Displacement units: counts, radians, degrees or distance units.	Displacement units: counts, radians, degrees or distance units.
Acceleration conversion value	
Initial position	Current position
Velocity	Velocity
Velocity range (max, min)	Nominal velocity
Acceleration	Acceleration
	Nominal acceleration
Homing type	Homing type
Motion type	Motion type
Negative and positive limits	Clockwise and counterclockwise limits
Active unit flag	Api status
Configured flag	
Maximum displacement	Full step enable
	Degrees per step
	Power in motion and standing
	Current
	Home ring offset
	I/O mapping

Table 8. Attributes techno_axis and api_axis classes.

In addition to the classes managing the properties of each axis, a second set of low-level classes (*api_functions* and *techno_functions*) provides the functionality for performing motion, input/output and configuration tasks.

The classes forming the different levels are described in greater detail in the following paragraphs.

The object of the *station* class, on its creation, manages the dynamic allocation of the memory space for the *api_axis* or *techno_axis* objects, one per axis in the system. The default number of axes is four (user modifiable) and they are arranged on an array so that the *station* object can easily address the appropriate axis for each action.

Each controller has a character identifier, which is initialized to NULL until the configuration file is read. The same happens with the active-unit flag, initialized to zero, or the communication port, whose default value is defined as a macro in the header of the *station* class.

The configuration file starts with the identifier of the serial port for communication, followed by the number of axes in the system, and the sequence of axis identifiers. The parameters for each axis can be for example arranged in columns (tab indent). The kind of configuration file we used for this project has the following design:

```
***** Serial port ******
%port        /dev/ttyS0
***** Number of controllers and their identifiers *****
%api_num     4
%apis        xywz
%activate            0        1        1        1
***** Home type: 1:switches, 2:ring, 3:hall , 4:hall (2)*****
%tipo_home    3  2        4        3
***** Motion type: 0:rotatory-linear, 1:rotatory-rotatory *****
%tipo_mov            0        1        0        0
***** Api managing CT relay*****
%rele_api
%rele_output
***** Api used for connecting handset *****
%handset_I_api    1
**** Api moving with handset orders *****
%handset_M_api   2
******** Groups id ********
%coldet_group 2

#EasySetUp_File    /motor_conf_files/owis.t.zip
    /motor_conf_files/newport_openloop_n2012.t.zip
/motor_conf_files/owis.t.zip
/motor_conf_files/owis.t.zip

********** Conversions ********************
1 pos_unit equals    mm              0.1deg          mm              mm
#rev_conversion     51200            2560            51200           51200
1 speed_unit equals...          mm/s    deg/s           mm/s            mm/s
#vel_conversion     40.96            20.48           40.96           40.96
#accel_conversion 0.03277            0.016384         0.03277         0.03277
********** Motion Variables **********
#axisposition                        0       0       0       0
#max_desp                            295     3600    95      295
#minvelocity                         3.0     5.0     3.0     3.0
#maxvelocity                         4.5     16.0    5.0     5.0
#axisacceleration            30      100.0   300     300
```

********** Inputs of interest **********

#lsn	24	0	33	33
#lsp	2	0	35	35
#home_input	34	38	34	34

Get and set functions are the body of these classes and allow easily handling and fetching of attributes of the particular axis.

As stated above, the functionality for performing motion commands is provided by another class, named *techno_function* and *api_func*, respectively. The controller module calls multiple methods from this class to program and execute any kind of displacement:

- Homing one or all axes.
- Moving an axis with specific speed, acceleration and/or distance.
- Waiting for one or all axes to complete the command.
- Interrupt a motion task in progress.

As in many other cases, the communication link needs to be opened when the system is initialized. Consequently, it must be closed when the system is shut down. Commands are passed on a write-read–like message interchange: *station* writes the order, *techno_functions* and *api_func* build and transmit the message, and lastly, the controller executes it and sends back an answer notifying the operation status. The *techno_functions* and *api_func* classes translate the requests by *station* into the commands understood by the particular hardware and forward them using the serial (RS232) channel.

For the case of the Technosoft controllers, they are directly accessed through a library provided by the manufacturer (*TML_lib* [33]).

On the other hand, when using API controllers, a dedicated software interface between the serial port and the user calls was to be implemented: the class *serial_io*.

Motion types

At first sight, the clearest classification of motion tasks differentiates between axis homing and single movements. *Home* functions return the selected axis, or all at once, to the position of reference while movement functions performs a concrete movement of the distance passed as input parameter.

Home routines vary depending on the type of axis. For the case of the linear stages (e.g. bed in figure 18), the task begins by stopping the axis wherever at the current position and reconfiguring the clockwise limit input as standard input. Continuous displacement starts until the limit switch is activated. For accuracy, the bed steps back at a lower speed until the

limit switch is deactivated and steps forward again at an even lower speed until it is activated again. This is called a hysteresis cycle.

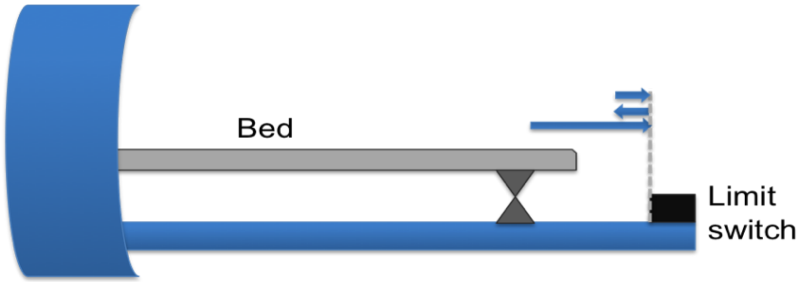


Figure 18. Schematic illustration of the homing process for linear axes.
The blue arrows illustrate the triple movement to verify it accurately reached home position.

The algorithm for *homing* the ring varies from one system to another. Nevertheless, we focus the explanation on the home algorithm for the fDOT-CT system since it is the system where our software was assessed. The homing process starts by moving the ring in the clockwise direction, until one switch (*mech*) placed at a fixed position changes its state. After detecting the change in the switch and depending on its original state and the distance traveled, several scenarios are possible:

1. If the state of the switch changes from 0 to 1 and the distance traveled was larger or equal than the distance separating the home position and the switch, denoted as *D* from now on, the ring starts moving in the counterclockwise direction a total distance of *D*.
2. If the state of the switch changes from 0 to 1 and the distance traveled was lower than *D*, the ring continues moving in the clockwise direction a total distance of 360º minus *D*.
3. If the state of the switch changes from 1 to 0, the ring continues moving in the clockwise direction a distance of 180º minus D.

In order to avoid hysteresis effects on the switch detection, the ring is moved back and forward more slowly to deactivate-activate *mech* a second time. The whole process is schematically depicted in figure 19.

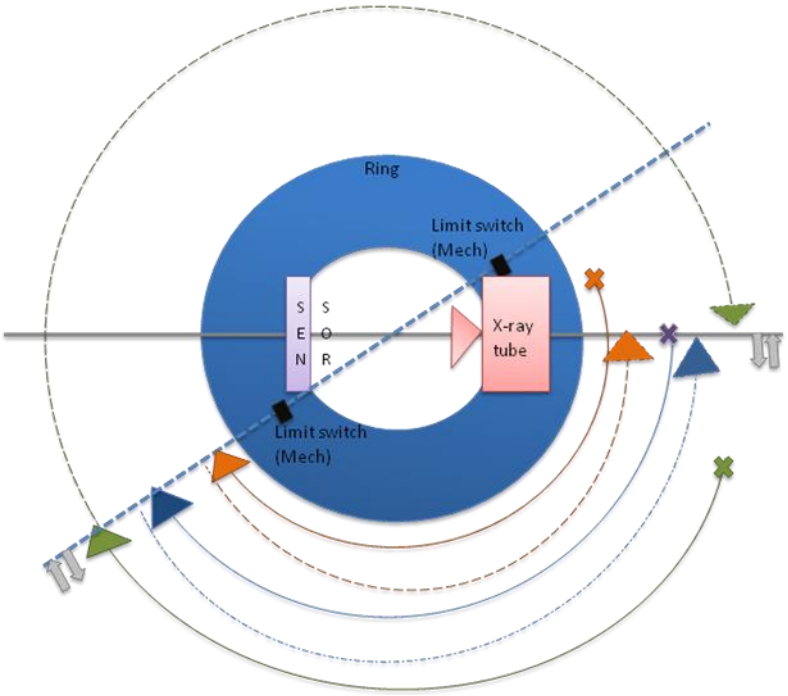


Figure 19. Ring home function for the fDOT-CT system. Crosses indicate the starting point, solid arrows the direction of the first displacement and dashed arrows the direction of the second one. Grey arrows illustrate the triple movement improve method accuracy.

The functions *moveaxis* or, for simultaneous motion of two axes, *moveaxes* can perform absolute, relative or continuous movement to the required position. The specified distance value has to be converted into controller internal units before transmitting the order. The difference in execution between the API and Technosoft controllers relays on the fact that motion commands are immediately executed in API systems, whereas in Technosoft systems, a first command programs the required movement and a second command starts the actual movement. If anything goes wrong, controllers send a command to immediately abort movement. Basic steps for moving an axis are summarized in the following flow diagrams in figure 20 (for API controllers) and figure 21 (for Technosoft controllers).

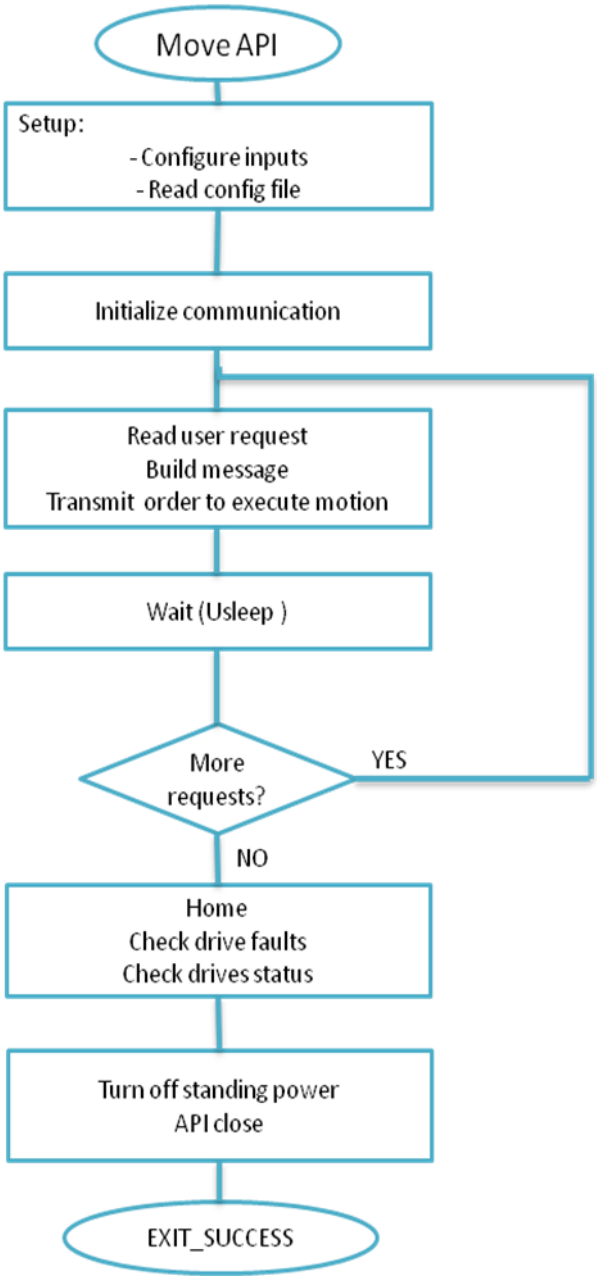


Figure 20. Flow chart illustrating the stages for performing a movement using API controllers.

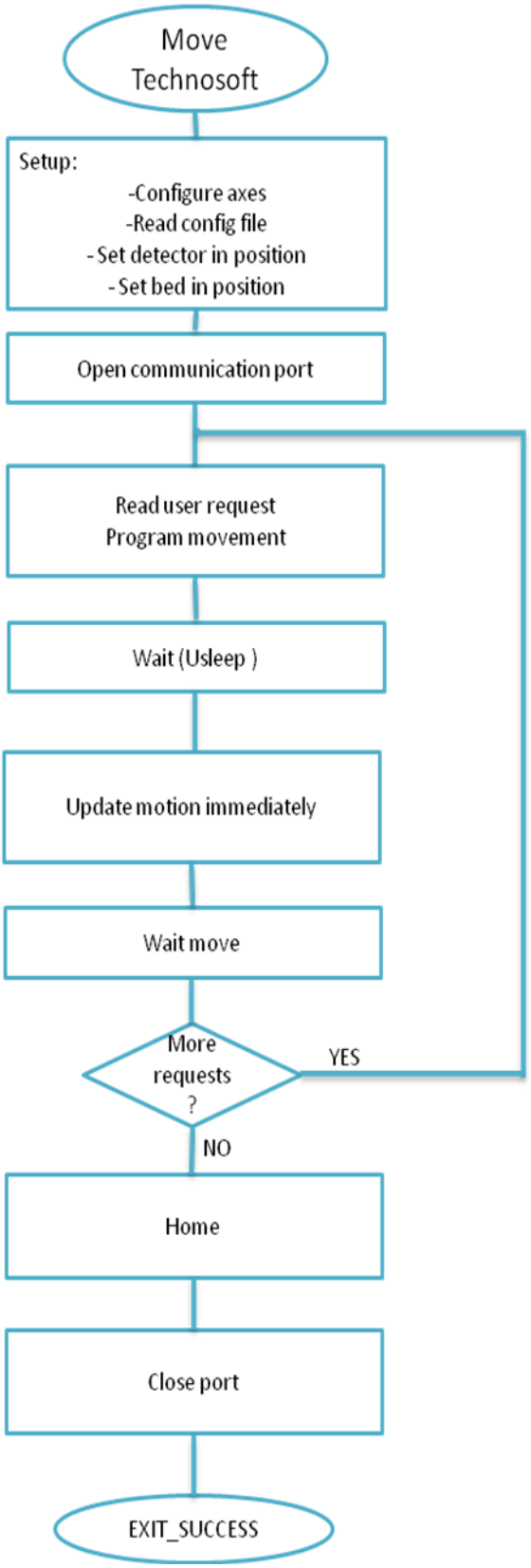


Figure 21. Flow chart illustrating the stages for performing motion commands for Technosoft controllers.

Table 9 summarized the main methods used when programming system movements.

Function	*Description*
Home	Homes the selected axis.
homeAll	Homes all the axes simultaneously.
Moveaxis	Programs and performs selected-axis motion for a specific distance.
wait_move	Waits until the selected axis stops moving.
wait_move_all	Waits until all *station* axes stop moving.
stopaxisImm	Aborts motion of the selected axis.
Power	Turns on/off the standing power of the selected motor.

Table 9. Motion functions.

4.4 X-ray source interface

Managing the x-ray source from the acquisition computer requires transmission of specific definitions and methods included in *tubemanger*, through the serial port. This instructions fall into one of the following categories:

- Control
- Monitoring
- Status notification
- Error notification

The control commands are not only the activate/deactivate pair but also the functions for setting up emitting flux voltage and current, performing self-test, resetting from an overload status or starting the warm-up process. Unless the warm up protocol has successfully finished, i.e. the tube is ready, these methods cannot be called.

The x-ray source has an internal watchdog timeout. If it goes off, the radiation is stopped since no command has arrived for a guard interval. Monitoring consists on checking the watchdog timeout value either periodically or on demand; and also on sending a command for resetting the watchdog.

The status of the source may be notified at three different points in time: before warming up, for retrieving hardware errors, interlock status and the pre-heat monitor; during the warming up process, to obtain data about the heating status; and once the system is ready, determining which tube is interacting, tube status or checking/updating the x-ray tube real output voltage and current. Emission time and internal battery requests are also available.

The error notification function allows interpreting the different error codes that could arise during any command execution.

Attributes of the *tubemanager* class store the configuration information of the source and they are accessible through get/set functions. Table 10 shows the tube structure.

struct SXTubeInfo
X-ray source model
Actual voltage
Actual current
Actual status
Warm up status
Warm up pattern
Warm up step
Interlock status
Hardware error
Watchdog timeout value
Power time
Emission time

Table 10. X-ray source information structure within tubemanager.

Tubemanager instances can be built using void or one-parameter constructor. The second option assumes the communication channel is already setup and ready to transmit and it receives as input parameter the file descriptor for the port. The control and status methods available in the *tubemanager* library are described in table 11. The structure handlers are not included as they are just accessories with get/set prototypes.

Function		***Description***
Control methods	**x_on**	Turns on x-ray emission.
	x_off	Turns off x-ray emission.
	x_setVC	Sets user-defined voltage and anode current.
	x_chWD	Checks the Watchdog timeout value
	x_tsf	Starts the self-test.
	x_rst	Resets from over load status.
	x_wup	Start the warm-up protocol.

Status methods	**x_checkConn**	Checks the communication link.
	x_typ	Discovers with which tube the system is dealing.
	x_sts	Queries the tube about its current status.
	x_swe	Warm-up process status check.
	x_sws	Warm-up process step check.
	x_snr	Checks the status when the tube is not ready.
	x_svc	Checks the real output voltage and current.
	x_update_set	Checks the preset voltage and current.
	x_check_times	Checks the soure power-on and emission times
	x_check_battery	Checks system internal battery status.
	x_err	Interprets error codes.

Table 11. *Tubemanager* methods.

4.5 System interface

The acquisition interface acts as the coordinator of the three modules: detector, controller and x-ray source. Among other tasks, it has to manage memory allocations, parse command line parameters and initialize each device with the correspondent configuration data.

A single method from this class is able to perform complex tasks that require multiple calls to the underlying libraries. The collection of methods included in the class and a brief description of their functionality is provided in table 12.

Function	***Description***
Parse_args	Read and store in a structure the values given as a result of the input parameters
Init_CT	System initialization with the values read from the system configuration file.
Init_det	Detector/frame grabber initialization.
Init_tube	X-ray source initialization.
Launch_WUP	Launch x-ray source warm-up protocol.
Alloc_shmem	Some acquisition strategies require two simultaneous processes running and accessing frame buffers. Therefore the shared resources must be concurrently handled.
Check_interlock	Check system interlock status. Called whenever the system is going to move.
Contolc	To abort execution saving all the necessary data and closing connections.

X-ray_turnon and x-ray_turnoff	Radiation control.
Set_bed and set_bed_home	Set the coach to the desired position. It receives as input parameter an integer indicating whether the watchdog must be reset.
Dark_acq and flood_acq	Acquire data with correction purpose.
Step_shoot	Step-&-shoot thread function.
Cont_shoot	Continuous shoot thread function.
Data_process	Data processing thread function.

Table 12. Description of *daq_sys* methods.

Constructor

The *daq_sys* class does not have any other constructor apart from the void constructor, mainly because any design parameter will be parsed from the program input arguments. Global variables belonging to this class are *detector*, *station* and *tubemanager* objects that have their own constructors, as it has been shown in previous sections of this chapter.

It is worth mentioning that *daq_sys* has a default set for the initialization of input arguments in case they are not specified at runtime. The assigned values are shown in table 13.

Acquisition parameter	Default value	Cmd option
Number of angular positions	360	-p
Detector binning (1,2 or 4)	4	-b
Frames per position [1…Inf]	1	-n
Image axial size (px)	Detector's maximum pixels in Y	-x
Image radial size (px)	Detector's maximum pixels in X	-y
Acquisition mode: 10=normal step-shoot 11=normal continuous	10	-s
Anode current (mA)	0.0 float	-a
X-ray source peak energy (kVp)	0.0 float	-v
Bed positions to scan	1	-z
Overlap between two consecutive bed positions (mm)	20	-c
Initial bed position (mm)	0	-d
Scan angular span to perform rotation: projection per scan angle/noi (deg)	360.0 float	-k

Source-to-detector distance (mm)	Detector dsd_default value	-m
External trigger through frame grabber: 0 NO - 1 YES	0	-e
Exposure time (ms)	0 (No external trigger, no need to select exposure time)	-e
Scan files base name	"unknown_patient"	-f

Table 13. Acquisition rutine default values.

Acquisition procedure

The scanner is ready to perform the acquisition tasks after each component has been initialized. Once the acquisition scheme is determined, the system starts scanning and collects the images that are stored for user processing. To shut down the system, memory has to be released and communication channels have to be closed.

The complete acquisition requires of two execution processes. One of the processes is in charge of carrying out the movement of the system while synchronizing it with the flat-panel detector. In the meantime the other process transfers the data provided by the detector from the shared memory buffers to user-defined memory/disk.

In the following sections we explain in detail the tasks involved in each of the stages.

Initializing x-ray source

Before starting the initialization process, it is necessary to check whether the x-ray tube is enabled. The first step is to setup the communication port and perform some preheating tests: communication link availability, source model and status, batch check, time properties. Then, assuming every returned value is correct, set the periodic Watchdog check.

At this point, the status of the x-ray source will correspond to one of these cases:

- Needs to be warmed up before performing any operation, so the warm up process (wup) function has to be called next.
- Already performing wup, status that will cause the acquisition to abort.
- Already emitting, status that also aborts the acquisition.
- Overload protection activated.
- System error or interlock is activated, i.e. the source is not ready.
- It is performing the self-test process.

Initializing the frame grabber and detector

Instantiating the detector object sets the default parameters, which may be modified according to the configuration file within the opening function.

Once created, enough memory must be allocated for the buffer used to store images in disc. The shared memory structure must be defined here as needs to allocate buffers and semaphores.

Initializing the motion system

The motion controller system *station* requires in its creation two axis objects, one for the bed and another one for the ring, and format file.

Once *station* is instantiated, a call to station configuration method configures the motion parameters of the axes. In order to unequivocally know the location of every mechanical component, axes should be homed. Thus subsequent moves can be performed referred to that initial point.

Acquisition loop

The user can select the acquisition scheme, together with the rest of acquisition parameters, by setting the proper flags and operands when launching the application.

System fetches dark data, or in other words, checks what is detected without exposure, just before turning the x-ray source on. Dark frames are used in correction stages later on.

Once both the gantry and the tube are ready, the acquisition protocol is loaded (step-and-shoot or continuous), in order to make it possible to obtain the projection data for the different angular positions.

Most of the current systems use a continuous acquisition scheme with continuous x-ray beam flux. Provided a fast detector is used, this approach achieves better acquisition times. Nonetheless, it may increase the effect of image lag and the number of mechanical imperfections and synchronization flaws.

A popular alternative is the acquisition of a stack of frames for each angular projection (figure 22), with the gantry steady during the acquisition (step-and-shoot). While the gantry rotates, no image is acquired and the radiation beam is blocked to avoid the delivery of useless dose to the subject under study and to minimize the effect of image lag.

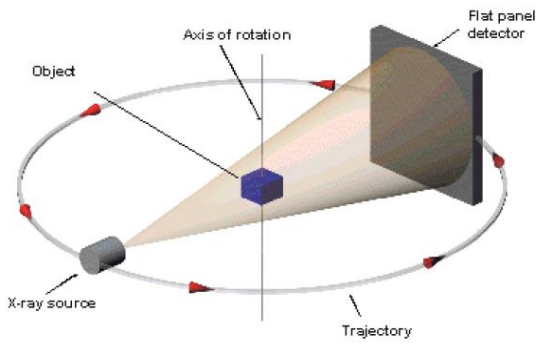
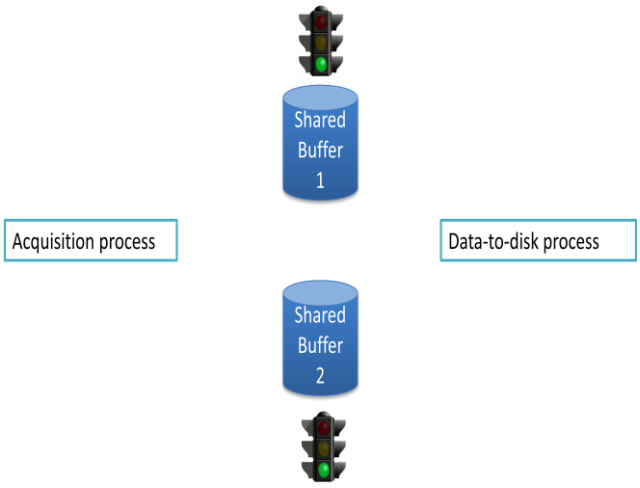


Figure 22. Angular projection depiction.

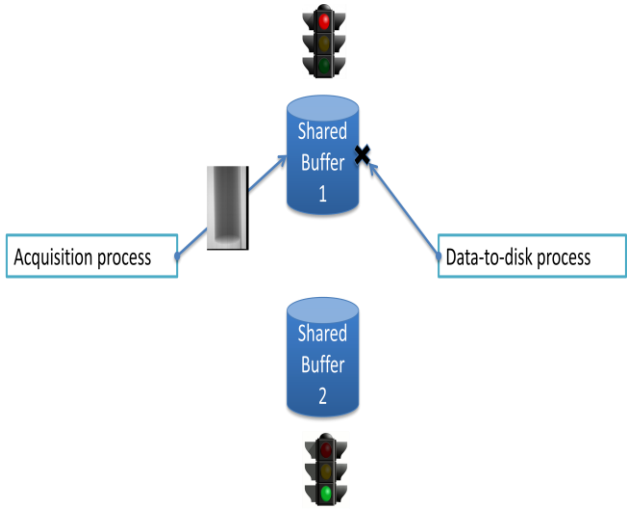
As it was mentioned at the beginning of this section, the acquisition interfaces requires of two executing processes. The processes were synchronized using semaphores and shared memory objects of BOOST::Interprocess library (Version 1.53.0) [34].

Deepen on the step-and-shoot approach, the acquisition process was implemented as an event-driven finite-state machine with states that take advantage of the maximum data rate provided by the detector [18], thus reducing acquisition time and delivered dose. Since we were interested in acquiring an averaged image for each angular position, we chose to acquire in sequence mode and to store the datasets in two shared memory areas in ping-pong style.

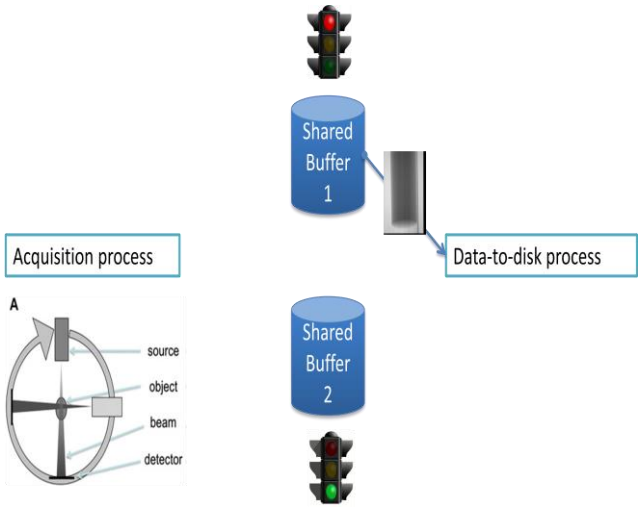
The following graphs provide an overview of the states of the system.



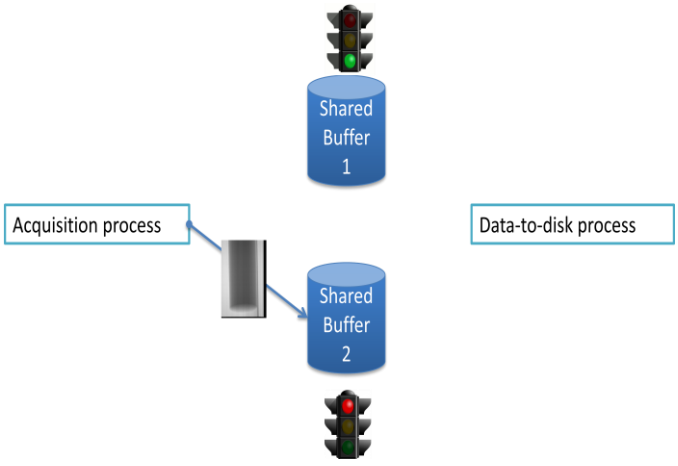
Steady state: Once the system is initialized, the shared memory objects are ready to be used: buffers have memory allocated, and mutex semaphores are unlocked. Each process uses a pointer to the current buffer.



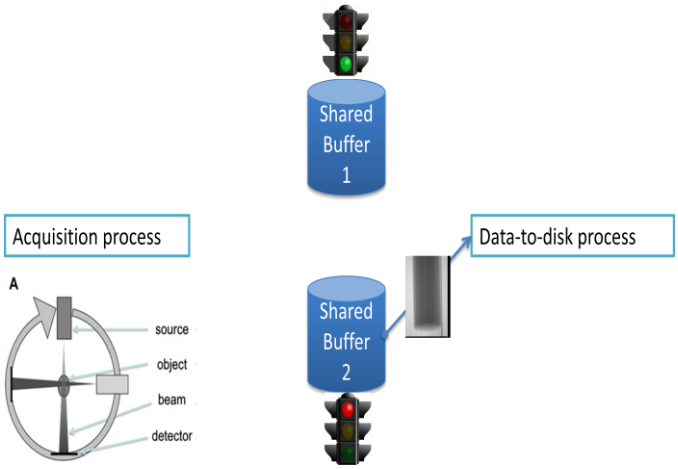
Shoot state: The acquisition process locks the mutex of the corresponding buffer (in this case buffer 1) and acquires the averaged image from the flat-panel at the current position. Then, it writes the data to the shared buffer, for which he has exclusive access –the data-to-disk process cannot read data yet–. Finally, it releases the shared memory area and updates its pointer to next buffer (buffer 2).



Step state: The acquisition process updates the gantry position. Meanwhile the data-to-disk process, after locking the semaphore, accesses the shared buffer where the dataset acquired at the previous gantry position has been stored (buffer 1). Once the data-to–disk process has finished reading the data, it releases the shared memory area and updates its pointer to next buffer (buffer 2). The data-to-disk process stores the datasets one after the other.



Shoot state: The machine returns to the shoot state, but this time accessing a different shared area. The behaviour of the processes is exactly the same as described in the first shoot state: the acquisition process acquires the image in the current gantry position, while the data-to-disk process waits for the dataset to be ready.



Step state: Analogue behaviour but processing the dataset from buffer 2 and moving the gantry forward to the next angular position. Once the data-to-disk process releases the shared area, the system switches to shoot state.

Figure 23 Diagrams of step-and-shoot acquisition schema.

The acquisition loop continues so on and so for until the system has gone through all the defined angular positions. After acquiring all the projection data, the data acquisition process turns the x-ray source off and homes the mechanical system; and the data-to-disk process dumps the images to a file and clears its buffer to be ready for new datasets.

System Finalization

The acquisition process closes both the motor system and serial communication link. Next call is the detector stopping function, which can be seen as part of the releasing memory procedure together with deallocating the global variables. Meanwhile, the data-to-disk process releases its allocated memory.

Now, the system is ready to exit the acquisition and data-to-disk processes.

Chapter 5

Assessment

Each coded module has its individual test bench, from the lowest level of the interface to the highest. The final data acquisition executable directly interacts with the acquisition interface, unaware of the underlying components integrating this platform. In this chapter, every stage of the followed testing protocol is described. The micro-CT models used for the assessment of flat panel sensor and controllers were the Argus system and the fDOT-CT system. The x-ray source and acquisition interfaces were tested on the fDOT-CT system.

5.1 Detector module

Test bench

With the implementation of three main global functions, we were able to run firstly a program that acquired a single snapshot and a sequence of 20 images (see figure 24); and secondly, to test the continuous acquisition approaches, we implemented a test program with two processes, one for image acquisition and a second one for image processing, as shown in figures 25 and 26.

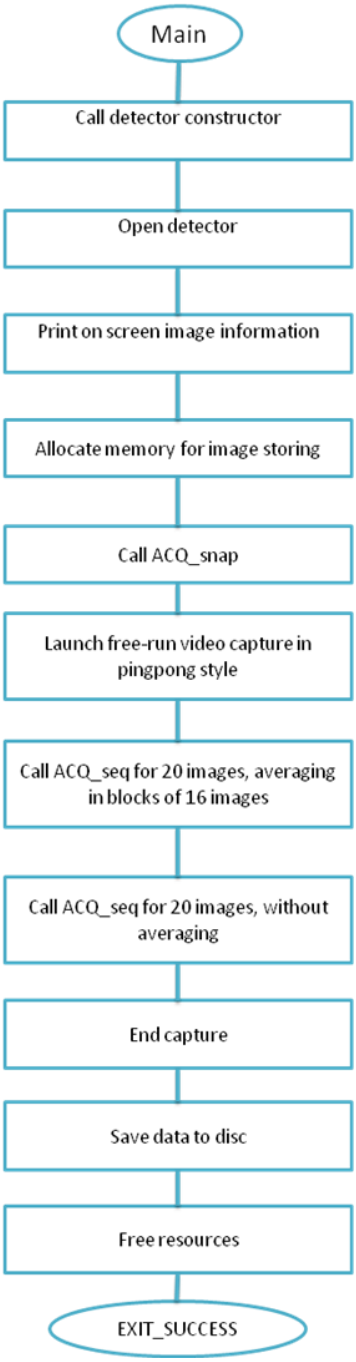


Figure 24. Flow diagram for snapshot and sequence acquisition

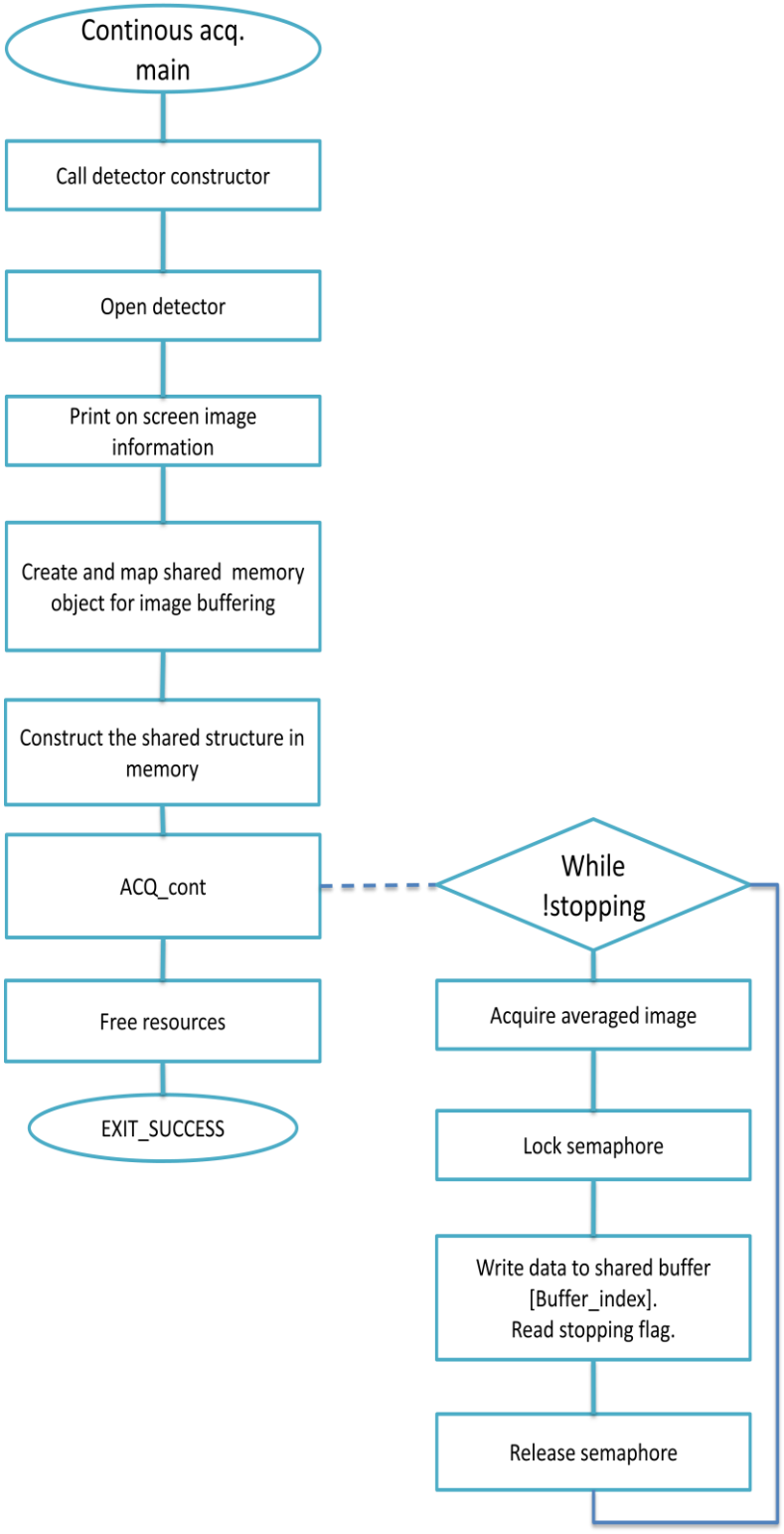


Figure 25. Flow diagram of the process managing the detector and frame grabber.

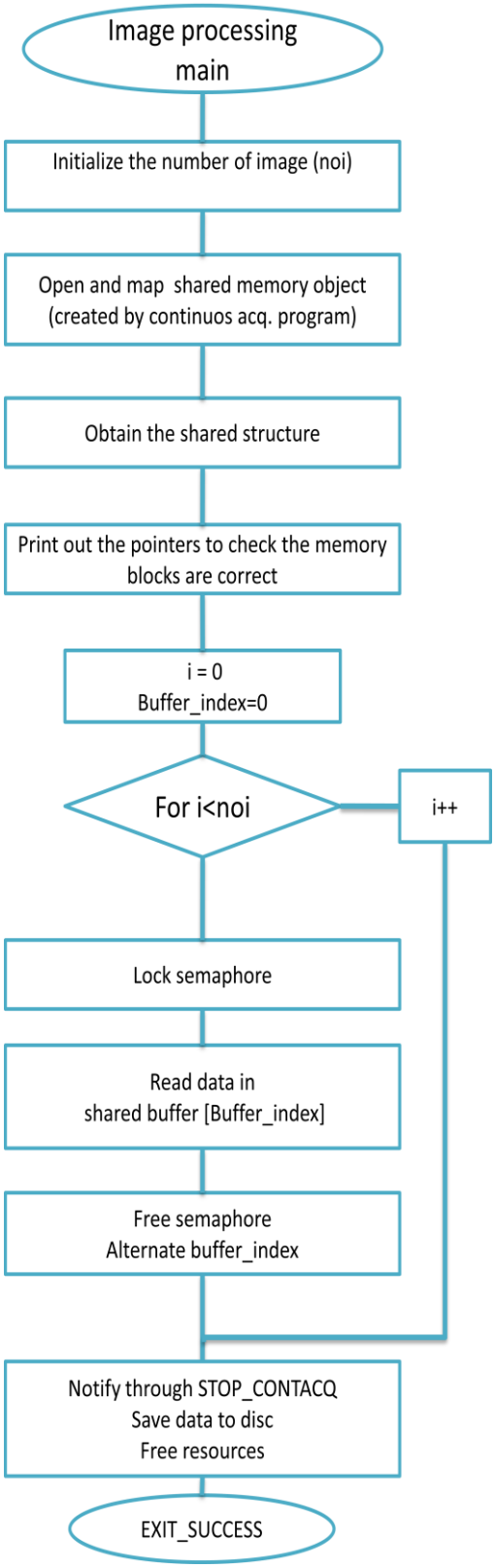


Figure 26. Flow diagram of the process dumping the acquired datasets.

5.2 Gantry module

Motor shell

To perform customized movements and have full control of the tests we implement a shell. The motor shell is a program that allows us to type in commands and run them in Unix system: controllers accomplish the request, and the motor shell shows on the screen the generated reply.

The supported input commands are enumerated in table 14.

Command synopsis	*Description*
Opencom	Initialization command. Opens communication serial port.
configapisystem configfile **config** configfile	Configures the motion system with the configuration file path passed in the first argument.
home axis_id	Homes the axis whose char identifier corresponds with the first argument.
get axis_id	Prints information about position, speed and acceleration.
relmove axis_id [-]distance	Moves the axis whose char identifier corresponds with the first argument, to the relative position indicated by the distance and direction of the second argument, assuming it is expressed in millimeters/degrees.
absmove axis_id [-]distance	Moves the axis whose char identifier corresponds with the first argument, to the absolute position indicated by the distance and direction of the second argument.
Exit	Exit the program.
close **closeAPIsystem**	Before leaving motor shell, the communication with controllers must be closed. It also turns off standing power of the motors.
kill **k**	System emergency stop. Requires system reconfiguration.

Table 14. Motor shell commands.

With these commands we could verify the behavior of the motion libraries: bed and ring displaced the specified distance, home position was reached accurately, hysteresis was minimized and motion could be precisely repeated.

5.3 X-ray tube module

We designed a set of tests for the evaluation of the individual features of the library managing the x-ray tube.

In order to minimize radiation and detector aging, functions devoted to turn on and off x-ray flux were not included in the assessment of the individual library and their validity was proved, using the system acquisition interface.

Test bench

A testing program was designed to query and configure the x-ray control unit, despite the restriction of not using XON/XOFF commands. The goal was to check that communication channel transmitted the desired requests by printing out the received responses.

With this in mind, we first setup the port and created a *tubemanger* object using the corresponding port descriptor. We proceeded to configure voltage and current and query about the x-ray source features and status, fulfilling the information sheet shown in table 15.

Values for tube structure:
Model: L9631MOD2
Preset voltage value: 60 kV
Preset current value: 90 μA
Real voltage value: 0 kV
Real current value: 0 μA
Current status value: NOT READY
HW Error: 0
Interlock status: 1
Power time: 962 h
X-ray emission total time: 11 h

Table 15. Tube information sheet.

5.4 Complete system

The previous sections introduced the dedicated tests aimed at the evaluation of individual libraries of the software architecture. However, they are meant to work together providing full functionality to the micro-CT system. Since the acquisition interface integrates all components under a single library, there were only missing two processes running independently, one for managing the system and the other one for processing the acquired data.

With all libraries and applications for the control of micro-CT system successfully running and cooperating, we planned a step-and-shoot acquisition of a soft-tissue rat-sized phantom

with cylindrical shape and 4 equiangularly spaced metallic threads placed longitudinally over its surface.

To assess the correctness of the step-and-shoot implementation we performed a preliminary test, simulating the whole acquisition process with no x-ray radiation. To this end, we included an option in the *daq_sys* command synopsis, which disables the x-ray source (--no_xray). This implies that tube related operations were not perform until the step-and-shoot scheme worked properly from start to end and images were correctly saved to disk.

The final test consisted in the acquisition of 360 projection images for a scan angular span of 360º with a 1º step, retrieving a 600x600 projection (binning 4) per angular position. To obtain an appropriate signal level at the detector we selected the working point, in terms of x-ray source voltage and beam filtration, that maximized the pixel signal for rat-sized phantoms. Figure 27 shows the plot showing the optimal voltage as a function of filtration for rat-sized samples, extracted from [22].

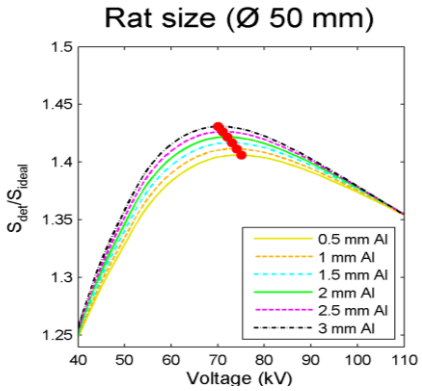


Figure 27 Relative signal compared to an ideal, flat-energy response
detector as a function of x-ray tube working voltage and xray beam alumninum filtration.

We used 66 kVp (+ 0.2 mm Cu) as optimal x-ray beam, inferred from experimental results on optimization of the acquisition protocol described in [22].

The anode current needed to get a suitable signal level (approx. 90 % of pixel saturation level) was 90 μA, yielding an output power of 6 W.

5.5 Results

With the acquisition protocol described in the previous section we were able to acquire a complete angular span of 360º of raw images. However, raw data required several correction steps before being suitable for tomographic image reconstruction. In this section, the resulting images are presented as raw data, corrected data and after tomographic reconstruction.

Raw data

The frames from the step-and-shoot acquisition for the assessment of the system interface were stored as 16-bit unsigned images of size 600x600 pixels with little-endian byte order. One of the acquired raw projection images can be seen in figure 28.

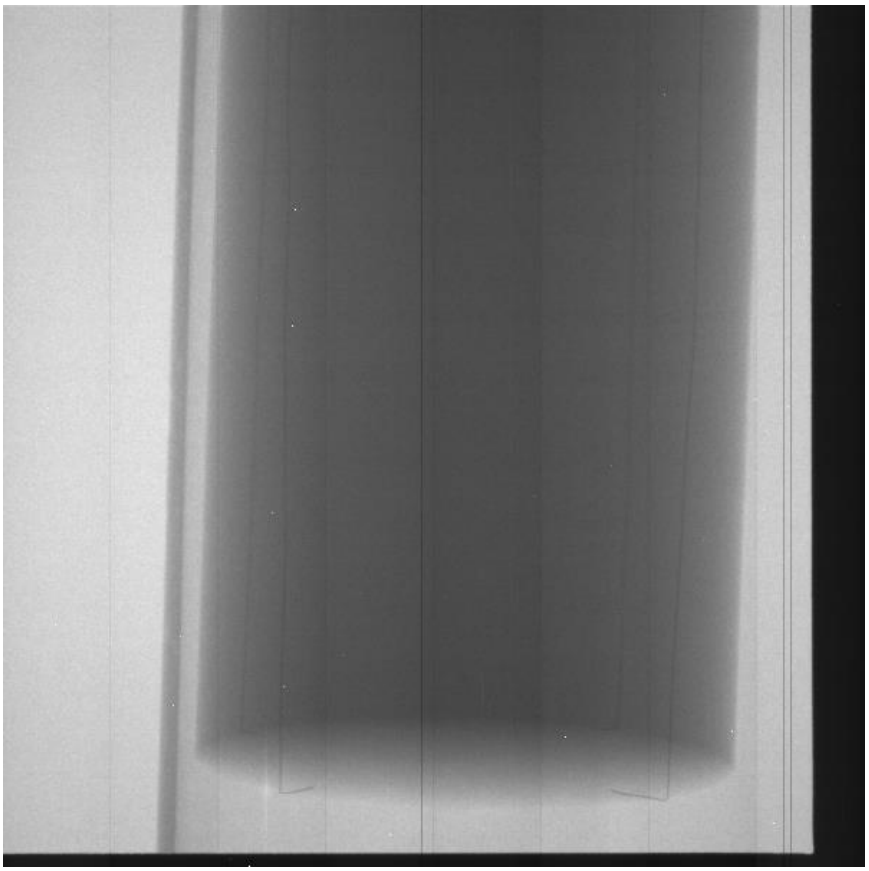


Figure 28. Projection acquired at 225º (66 kV - 90 μA).

As stated above, this raw image contains several non-idealities that must be addressed prior to tomographic reconstruction:

- By design, the active area of the detector is smaller than the physical area.
- The gain non-uniformity of the pixel response creates structured noise that shows up as a bumper-like pattern.
- There are some faulty rows, columns and individual pixels.

The methods used to compensate these non-idealities are described in the next section.

Processing

Inactive detector areas were removed from the processed projection data by cropping the original images.

After reducing the image area to the active part, defective elements were corrected by substituting their original value with the interpolation of the values of the surrounding elements.

There are several approaches to compensate the remaining non-uniformity in pixel gain, among which we have selected the classic flat-field and dark current removal for the correction of raw data in the presented scanner. It has been shown to largely compensate the non-uniformity of pixel response and it is easy to implement as a Matlab script. For these corrections we acquired the flood and dark images shown in figure 29.

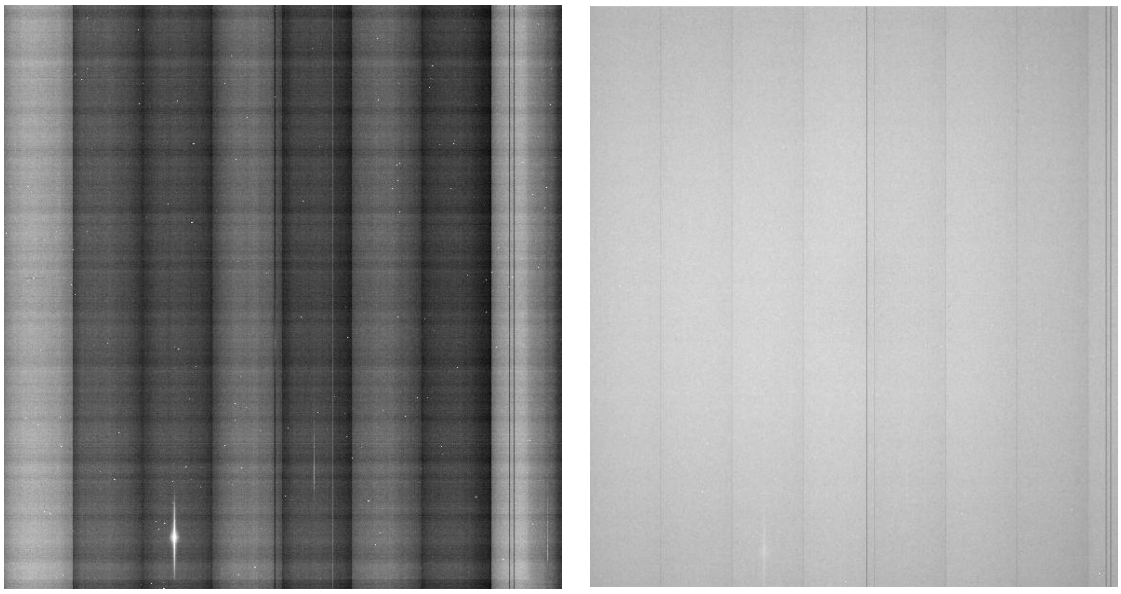


Figure 29. Dark map (left) and flood map (right) for the same exposure settings (66 kV 90 μA). Acquired prior step-and-shoot projections.

The pixel value, after subtracting the dark offset, is divided by the result of subtracting the dark offset to the flood data.

The results of this procedure can be assessed by the comparison shown below, in figure 30.

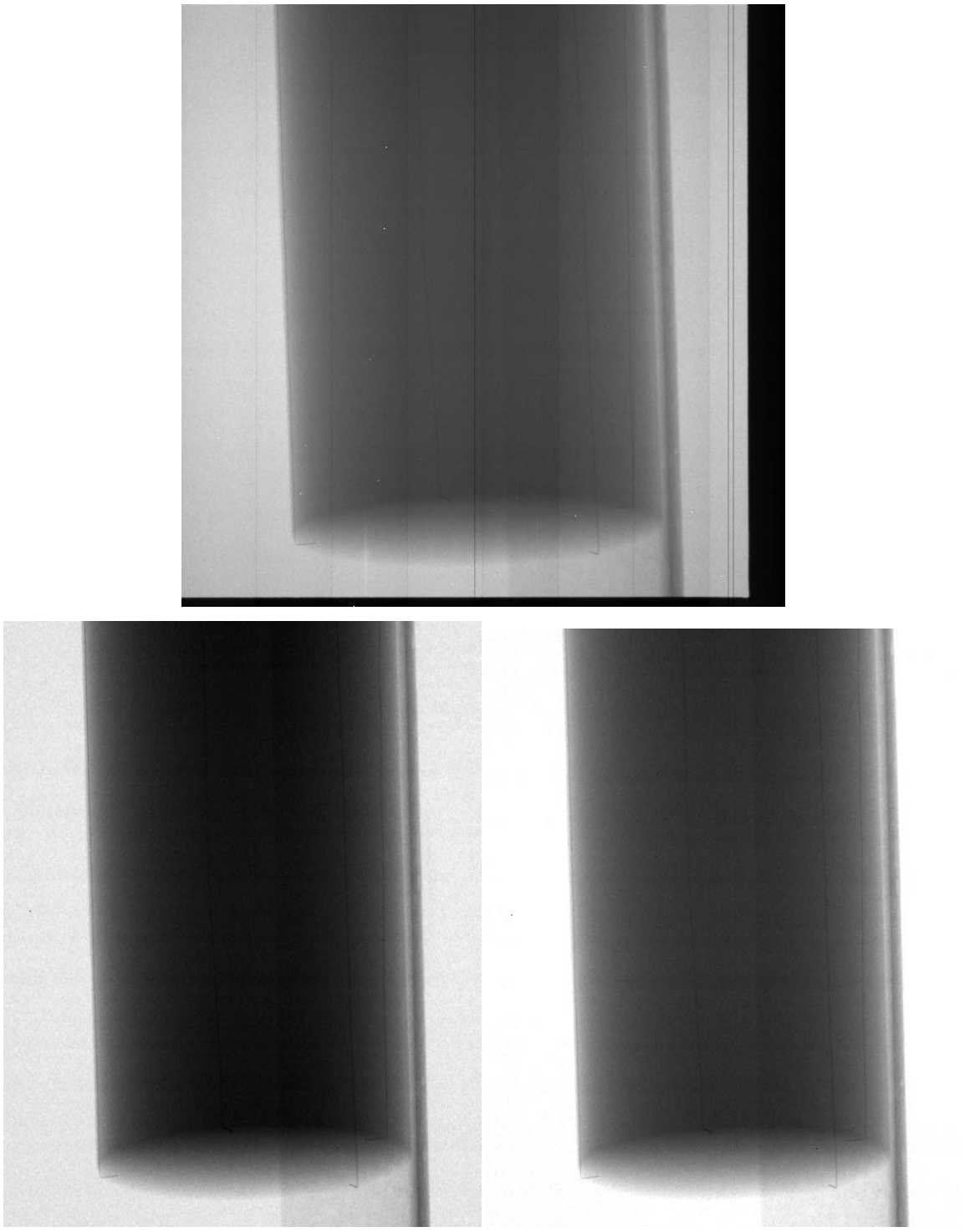


Figure 30. Raw data (top row) and processed data (bottom row) with two different contrast/brightness adjustment.

Image reconstruction

For the reconstruction of tomographic data we used a filtered back projection-like approach implemented as part of [22, 35]. This approach is based on the well-known FDK algorithm [15]. A frequency ramp-like filter, yielding the high frequency version of the original projection data, filters every row of the pre-processed projection. The filtered projection is then backprojected along its angular direction onto the voxel grid of the reconstructed data.

These three steps must be carried out sequentially, since the output of each one of them is the input for the next, but every pixel can be processed independently within each step. This is the reason why the software takes profit of the highly parallelizable nature of the reconstruction process and the massive parallel computation capabilities offered by Graphic Processing Units (GPUs).

For the present acquisition we performed two reconstructions of 560x560x586 voxels: one using a Shepp-Logan filter and another one using a Hann filter. For both images the cut-off frequency was set to the Nyquist frequency.

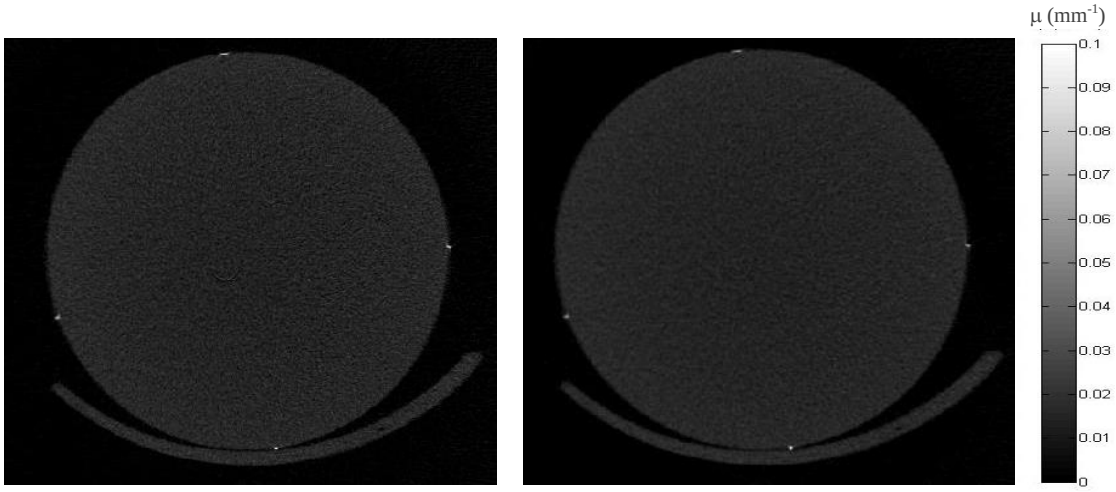


Figure 31. Reconstructed image with a Sheep-Logan (left) and a Hann (right) filter.
The color bar on the right of both images represents the attenuation values.

The degree of cupping in the reconstructed images gives a hint about the uniformity of attenuation values across the volume. For its estimation, we measured the mean value of an 18x18 voxel ROI at the centre and close to the edge of the cylinder, defining μ_{center} and μ_{edge} respectively, and computed cupping using the following expression

$$t_{cup} = 100\,x\,\frac{\mu_{edge} - \mu_{center}}{\mu_{edge}}$$

Results yielded a cupping level of 12.5 %, in agreement with results in the literature [22] for the same system, proving the correctness of the data provided by our new software implementation.

Chapter 6

Project Budget

UNIVERSIDAD CARLOS III DE MADRID
Escuela Politécnica Superior

PRESUPUESTO DE PROYECTO

1.- Autor:
Ana Ortega Gil

2.- Departamento:
Bioingeniería e Ingeniería Aeroespacial

3.- Descripción del Proyecto:

- Titulo	**Desarrollo software de librerias y aplicaciones para el control micro-CT**
- Duración (meses)	**8**
Tasa de costes Indirectos:	**20%**

4.- Presupuesto total del Proyecto (valores en Euros):
44.000,00 Euros

5.- Desglose presupuestario (costes directos)

PERSONAL

Apellidos y nombre	N.I.F. (no rellenar - solo a titulo informativo)	Categoría	Dedicación (hombres mes) [a]	Coste hombre mes	Coste (Euro)	Firma de conformidad
Alejandro Sisniega		Ingeniero Senior	4	4.289,54	17.158,16	
Ana Ortega		Ingeniero	4	2.694,39	10.777,56	
		Hombres mes	**8**	**Total**	27.935,72	

[a] 1 Hombre mes = 131,25 horas. Máximo anual de dedicación de 12 hombres mes (1575 horas)
Máximo anual para PDI de la Universidad Carlos III de Madrid de 8,8 hombres mes (1.155 horas)

EQUIPOS

Descripción	Coste (Euro)	% Uso dedicado proyecto	Dedicación (meses)	Periodo de depreciación	Coste imputable [d]
Dell OptiPlex 745	1.500,00	100	8	60	200,00
Matlab License	6.000,00	100	2	60	200,00
				Total	400,00

[d] Fórmula de cálculo de la Amortización:

$$\frac{A}{B} x C x D$$

A = nº de meses desde la fecha de facturación en que el equipo es utilizado
B = periodo de depreciación (60 meses)
C = coste del equipo (sin IVA)
D = % del uso que se dedica al proyecto (habitualmente 100%)

OTROS COSTES DIRECTOS DEL PROYECTO[e]

Descripción	Empresa	Costes imputable
Internet	Telefónica	960,00
Fedora 9	Red Hat, Inc	0,00
Windows XP Professional x64	Microsoft	130,00
Office 2003	Microsoft	100,00
Putty	Simon Tatham	0,00
Image J	NIH Image	0,00
VNC	Real VNC	0,00
Material de oficina		150,00
	Total	1.340,00

[e] Este capítulo de gastos incluye todos los gastos no contemplados en los conceptos anteriores, por ejemplo: fungible, viajes y dietas, otros,...

6.- Gastos generales y beneficio industrial

Sobre coste directo de personal	1.724,41	**16%**

7.- Resumen de costes

Presupuesto Costes Totales	**Presupuesto Costes Totales**
Personal	27.936
Amortización	400
Costes de funcionamiento	1.340
Costes Indirectos	5.935
Total sin IVA	35.611
IVA 21%	7478
Total estimado	43.089

Discussion and future work

The work presented in this project contributed to the development of the new version of the libraries used for acquisition and control of cone beam micro-CT systems and to the implementation of a transparent and usable interface for generic micro-CT devices.

This software is implemented in C++, which is upwards compatible with C and one of the most frequently used open languages with a wide range of compilers. Its object-oriented design compasses low level libraries in a single interface, preventing the high level applications from dealing with direct hardware interactions and isolating device drivers code (that really needs access to the hardware) from the complex acquisition schemes making use of them.

The aforementioned factors made this acquisition middleware a highly portable interface. Performance has been validated on the small-animal fDOT-CT system designed by J. Aguirre and A. Sisniega [35].

The low level libraries have also been integrated in a user-friendly acquisition interface in the platform LabView (see figure 32) that it is indeed internally used in the department.

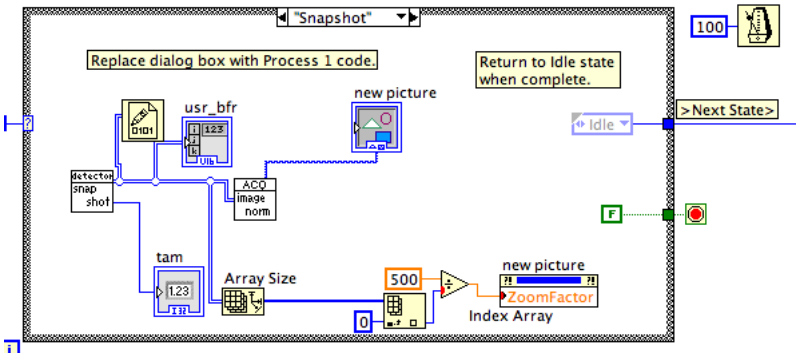


Figure 32. LabView snapshot virtual instrument. The snapshot vi is a call to the method in detector class.

Nevertheless, there are still some features that we will like to improve and to incorporate in the interface. Although the continuous mode has been implemented, it has not been evaluated yet. Neither has the instruction that performs the flat-field calibration acquisition, since calibration data is currently acquired by an independent process and not as part of the micro-CT data acquisition process.

As the acquisition scheme gets more complex, interacting with the encoders for retrieving the exact position of mechanical components will grow in importance. Also, it will be worth to discuss the benefits of including correction algorithms at this hardware level or delegating the correcting tasks to higher-level applications.

The library for controlling the x-ray source from Oxford Instruments (used in the Argus CT system) will be implemented in the near future, so that the interface can be integrated in the Argus system used in the department.

Regarding its commercial use, this software will be included in at least two more micro-CT systems that are being manufactured by SEDECAL [7]. The first one, already mentioned in chapter 3, is the test bench for the biomedical laboratory in Carlos III University.

This design will be used by students in their laboratory practices for learning how micro-CT system works and for research purposes. It is a system that allows plenty of geometrical configurations thanks to the multiple linear and rotational axes, as it can be seen in the prototype in figure 33.

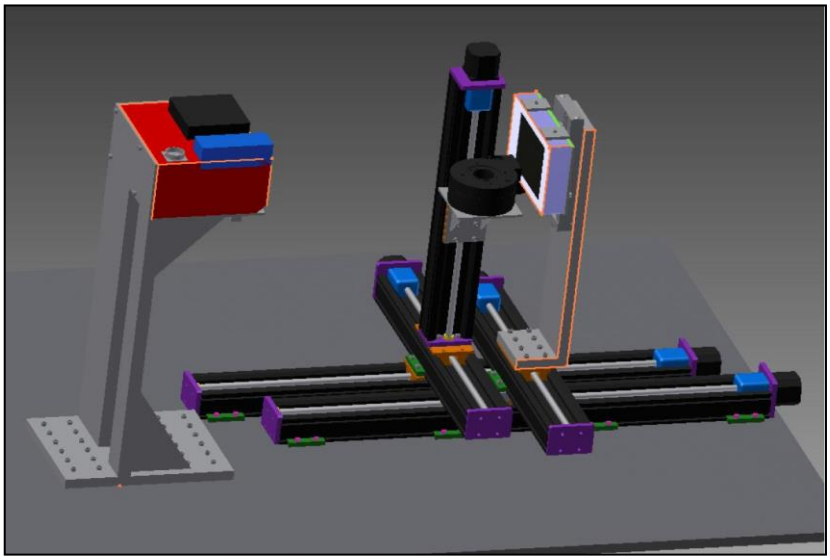


Figure 33. UC3M Test bench prototype.

The second system is one of the targets defined in the work project “Improving imaging platforms” for the IMI-funded PreDiCT-TB project, a research on model-based preclinical development of anti-tuberculosis drug combinations. One of the working packages includes the design and implementation of an in-vivo small-animal CT imaging with enhanced soft-tissue contrast that will make use of the presented libraries.

Annex A

System initialization log when simulating step-and-shoot acquisition

The log below shows the messages printed out by the software while the initialization of the system is taking place. Remember that before launching any acquisition protocol, the system must be initialized as it was explained in section 5.4, so this will be printed out when the acquisition system is started for the first time or after abort an acquisition in progress. The values are the ones read from the configuration files passed through command line in the real case scenario.

****

VERSION ACQ: 5.0

Subversions:

Daq CT Library version: 2.0

Motion Control Version: 1.0

X-ray Tube Management Version: 0.1 with Serial I/O Version: 0.1

X-ray Flat-panel Management Version: 1.0

Annex A System initialization log when simulating step-and-shoot acquisition

****

[init_CT] Opening CT configuration file...OK
[init_CT] File CT_params.txt was read.

**********CT configuration summary**********

Physical Args

HOME_X_POS: 5
HOME_Y_POS: 0
RING_START_POSITION: 0.000000
BED_STOP_MAX: 0
BED_STOP_MIN: -471220
JOY_VEL: 300000
BEDOFFSET: 335
PIXELS_X: 2400
PIXELS_Y: 2400
MAX_PIXELS_X: 2240
MAX_PIXELS_Y: 2344
PIXEL_SIZE: 0.050000
FILTER THICKNESS: 1.000000
FILTER MATERIAL: AL
ANGLE_OFFSET: 0.000000
ROT_DIRECTION: 1
CONTROLLER UNIT ID FOR RING: y
CONTROLLER UNIT ID FOR BED: z
CONTROLLER UNIT ID FOR DETECTOR: w
MINIMUM Dsd: 205.000000
DEFAULT Dsd: 205.000000
DET_TRAVEL: 95.000000
DET_STEP: 5.000000

Frame grabber configuration files

FORMAT_BIN_1: video_config
FORMAT_BIN_2: fconfig_bin2
FORMAT_BIN_4: fconfig_bin4
System correction and configuration files

D_CORRECTIONS_FILE: d_corrections_CT.txt
MOTOR_CONFIG: FMT-CT.txt
DARK_FILE_BASE: dark
FLOOD_FILE_BASE: flood

Swept parametres

FORMAT_SWEPT: fconfig_bin4
VOLT_SWEPT: 40.000000
AMP_SWEPT: 20.000000
OVERLAP_SWEPT: 0

Connection ports

MOTOR PORT: /dev/ttySNX1

[configureTECHNO] Initializing TECHNOSOFT

System on port/dev/ttySNX1
Baud Rate of 9600
HOST ID 0

[station::config] Station ready

TUBE PORT: /dev/ttyS0

Calibration Args:

GAIN_CAL_PATH: gain_cal_data
NUM_STEP_GAIN: 8
NUM_IMG_B4: 40
NUM_IMG_B2: 10

Annex A System initialization log when simulating step-and-shoot acquisition

NUM_IMG_B1: 10
GEOM_CAL_PATH: geom_cal_data

X-ray tube maximum ratings

MAX_CURRENT: 800
MAX_VOLT: 110

**********End of configuration summary**********

[parse_args] X-ray source disabled
[parse_args] Correction of projection images disabled

[init_tube] X-ray source disabled. Nothing to do

[init_det] Initializing values for X-ray detector
[init_det] Opening frame grabber device

Configuration file for frame grabber fconfig_bin4
Opening EPIX(R) PIXCI(R) Imaging Board
Using configuration param
Library version: PIXCI(R) 32 Bit Library 3.07.00 [09.11.04]
Driver version : PIXCI(R) 32 Bit Driver V3.7.002.6.27.25-78.2.56.fc9.i686
Image frame buffer memory size: 356352KBytes
Image frame buffers : 493
Number of boards : 1
Image resolution:
Xdim =600
Ydim =600
Colors = 1
Bits per pixel= 12

Annex B

System procedure log when user aborts acquisition

The user may decide to abort the acquisition for many reasons: an unexpected event occurred, the user forgot to set some parameters, something happened with the patient … Whenever something goes wrong, the command contol+c will abort the current acquisition. The following log report illustrates the steps for terminating the acquisition without leaving zombie processes or blocks of allocated memory not released.

[controlc] The user has aborted the acq process, exiting
[controlc] Shutting down x-ray source...
[xray_turnoff] X-ray source disabled. Nothing to do
[controlc] Stopping ring...
[quit_power] Switching off ...
[power] Power turned 0
[controlc] Stopping detector...
[quit_power] Switching off ...
[power] Power turned 0
[controlc] Stopping bed...

Annex B System procedure log when user aborts acquisition

[quit_power] Switching off ...

[power] Power turned 0

[close] El puerto de comunicaciones esta cerrado

[controlc] Comunication link closed

Stopping acquisition ...

PIXCI(R) imaging board closed

References

1. J. Radon, Über die Bestimmung von Funktionen durch ihre Integralwerte längs gewisser Mannigfaltigkeiten, in Ber. Verh. Sächs. Akad. Wiss. Leipzig, Math. 1917.
2. W.A. Kalender, *Computed Tomography*. 2005: Erlangen: Publicis Corporate Publishing.
3. *Interim Policy on Genomics*, E.E.P. Agency, Editor 2002, US: Science Policy Council.
4. W.T. Astbury, Molecular Biology or Ultrastructural Biology. Nature, 1961. 190.
5. S.C. Cauter, *Now you can image a mouse like a man.* Laboratory News (M.Sc.), 2005.
6. SCANCO. *Specimen microCT Brochures. SCANCO Medical AG.* 07/25/2013]; http://www.scanco.ch/fileadmin/webmaster_img/Brochures/XtremeCT_v3.pdf.
7. SEDECAL. *Super Argus PET/CT - SEDECAL*. 07/25/2013]; http://pmod.jp/pdf/08sedecal/supera-rgus-datasheet.pdf.
8. GE Healthcare, Technical Pubication GE Healthcare eXplore Locus User Guide. Vers. 2. 07/25/2013]; http://www.oucom.ohiou.edu/ou-microct/Downloads/eXplore%20Locus%20User%20Guide%20Rev%202.pdf].

9. Siemens. *Inveon PET-SPECT-CT Siemens Healthcare*. 07/25/2013]; https://http://www.medical.siemens.com/webapp/wcs/stores/servlet/PSOptionProductDisplayView?catalogId=-11&catTree=100010,1007660,1011525,1029715,1029721&langId=-11&productId=200641&storeId=10001].

10. J. Hsieh, *Computed Tomography*. 2009, Bellingham, Washington: SPIE.

11. A. Molins, Aquisición y Procesado de Datos en un Sistema de Imagen Animal Multimodal. 2004, Madrid: HGGM.

12. L. Juang, X-ray chest image reconstruction by Radon transform simulation with fan-beam geometry. Measurement, 2010. 43(3): p. 447-453.

13. J.A. Fessler, Statistical image reconstruction methods for transmission tomography, in Handbook of Medical Imaging, M.S.a.J.M. Fitzpatrick, Editor. 2000, ed Bellingham SPIE.

14. X. Jia, et al., GPU-based iterative cone-beam CT reconstruction using tight frame regularization. Phys Med Biol, 2011. 56**,** 3787.

15. L.A. Feldkamp, L.C. Davis, and J.W. Kress, *Practical Cone-Beam Algorithm.* Journal of the Optical Society of America A-Optics Image Science and Vision, 1984. 1612-619.

16. H. Illers, D. Vandenbroucke, and E. Buhr, Measurement of correlated noise in images of computed radiography systems and its influence on the detective quantum efficiency. Proc. SPIE, 2004. 5368(639).

17. S.M. Sze and M.K. Lee, *Semiconductor Devices: Physics and Technology*. 2012: Hoboken, N.J. Wiley.

18. Hamamatsu Photonics, K.K., Hamamatsu Application Manual, X-ray flat panel sensor C7912 & C7942 & C7943, rev 2.10, kr1-i50006, 2003.

19. A.C. Konstantinidis, Evaluation of digital x-ray detectors for medical imaging applications, in PhD Department Medical Physics and Bioengineering 2011, University College London London.

20. C. Schmidgunst, D. Ritter, and E. Lang, Calibration model of a dual gain flat panel detector for 2D and 3D x-ray imaging. Med Phys, 2007. **34**(3649-64).

21. J.A. Seibert, J.M. Boone, and K.K. Lindfors, *Flat-field correction technique for digital detectors.* Proc. SPIE, 1998. 3336(348).

22. A. Sisniega, Contributions to the improvement of image quality in CBCT and CBμCT and application in the development of a CBμCT system, in Bioingeniería e ingeniería aeroespacial 2013, Universidad Carlos III: Madrid.

23. A. Sisniega, J.J. Vaquero, and M. Desco, Design and Assessment Principles of Semiconductor Flat-Panel Detector-Based X-Ray Micro-CT Systems for Small-Animal Imaging, in Integrated Microsystems: Electronics, Photonics, and Biotechnology, E. K. Iniewski, ed New York: CRC Press, Editor. 2012. p. 309-335.

24. Micro Photonics Inc, *SkyScan 1176 Micro-CT*. 07/25/2013]; http://www.microphotonics.com/in-vivo-skyscan-1176-micro-ct].

25. CT Imaging, *CT Imaging - TomoScope (R) Synergy - Specifications*. 07/25/2013]; http://www.ct-imaging.de/en/ct-systeme-e/mikro-ct-e.html.

26. A. Sisniega, et al., Comparative study of two flat- panel X-ray detectors applied to small-animal imaging cone-beam micro-CT, in IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC)2008. p. 3836-3840.

27. H. Mori, et al., High resolution and high sensitivity CMOS panel sensors for x-ray, in IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC)2002. p. 29-33.

28. A.L. Goertzen, Development of a Combined microPET and microCT System for Mouse Imaging, 2003, University of California: Los Angeles.

29. Dexela Lmt., 2315 CMOS x-ray detector specifications, 2010.

30. Technosoft, S.A., ISCM4805/8005 v1.3 Technical Reference. 2010.

31. API, DRVKERNEL Serial Communications Kernel for the API Intelligent Microstep Motor Drives (DM-224i and DM-2205i). Vol. Version 2.1. 1999.

32. PIXCI, Reference manual XCLIBtm Imaging Board 'C' Library VERSI0N 3.7 2011: Epix, Inc.

33. Technosoft, S.A., TML_LIB v2.0 User Manual P091.040.v20.UM.0609. 2009.

34. I. Gaztanaga, *Chapter 13. Boost.Interprocess*. 2012 07/25/2013]; http://www.boost.org/doc/libs/1_53_0/doc/html/interprocess.html.

35. J. Aguirre, et al., *Design and development of a co-planar fluorescence and X-ray tomograph.* IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC), 2008: p. 5412-5413.